이학 박사 학위논문

Mathematical Analysis of the Indistinguishability Obfuscations

(구분불가능한 난독화의 수학적분석에 관한 연구)

2020년 2월

서울대학교 대학원

수리과학부

김지승

Mathematical Analysis of the Indistinguishability Obfuscations

(구분불가능한 난독화의 수학적분석에 관한 연구)

지도교수 천정희

이 논문을 이학 박사 학위논문으로 제출함

2019년 10월

서울대학교 대학원

수리과학부

김지승

김지승의 이학 박사 학위논문을 인준함

2019년 12월

위 원 장	김 명 환	(인)
부 위 원 장	천 정 희	(인)
위 원	이 향 숙	(인)
위 원	현 동 훈	(인)
위 원	윤 아 람	(인)

Mathematical Analysis of the Indistinguishability Obfuscations

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Jiseung Kim
Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

February 2020

Abstract

Mathematical Analysis of the Indistinguishability Obfuscations

Jiseung Kim

Department of Mathematical Sciences

The Graduate School

Seoul National University

Indistinguishability obfuscation (iO) is a weak notion of the program obfuscation which requires that if two functionally equivalent circuits are given, their obfuscated programs are indistinguishable. The existence of iO implies numerous cryptographic primitives such as multilinear map, functional encryption, non interactive multi-party key exchange. In general, many iO schemes are based on branching programs, and candidates of multilinear maps represented by GGH13, CLT13 and GGH15.

In this thesis, we present cryptanalyses of branching program based iO over multilinear maps GGH13 and GGH15. First, we propose cryptanalyses of all existing branching program based iO schemes over GGH13 for all recommended parameter settings. To achieve this, we introduce two novel techniques, 'program converting' using NTRU-solver and 'matrix zeroizing', which can be applied to a wide range of obfuscation constructions. We then show that there exists polynomial time reduction from the NTRU problem to all known branching program based iO over GGH13.

Moreover, we propose a new attack on iO based on GGH15 which exploits statistical properties rather than algebraic approaches. We apply our attack to recent two obfuscations called CVW and BGMZ obfuscations. Thus, we break the CVW obfuscation under the current parameter setup, and show that algebraic security model of BGMZ obfuscation is not enough to achieve ideal security. We show that our attack is lying outside of the algebraic security model by presenting some parameters not captured by the proof of the model.

Key words: Cryptanalysis, Indistinguishability Obfuscation, Multilinear Map
Student Number: 2014-21202

Contents

Chapter 1

Introduction

Intuitively, the program obfuscation is similar to an encryption scheme which takes as input a program, not a message. Informally, the security of the program obfuscation is to hide all information excepts for inputs and outputs of the program. Constructing a general-purpose program obfuscation has been a long standing coveted open problem because of fruitful applications and implications, but the impossibility of the general-purpose program obfuscation was proved [BGI$^+$01, BGI$^+$12]. Instead, authors of the seminal paper proposed a weak notion of program obfuscation, called the indistinguishability obfuscation. Currently, a cryptographic obfuscation means the indistinguishability obfuscation.

1.1 Indistinguishability Obfuscation

Indistinguishability Obfuscation (iO) is a weak notion of program obfuscation. It takes as input a program, and outputs a obfuscated program while preserving the functionality. The purpose of iO is to hide one bit informa-

tion which one of program is obfuscated when two functionally equivalent programs and an obfuscated program of one of them are given. Although it provides one bit indistinguishability, it has numerous applications such as a functional encryption [GGH+13b], a witness encryption [GGSW13], a deniable encryption [SW14], graded encoding schemes [FHHL18], and a traitor tracing [BZ17].

Garg *et al.* [GGH+13b] first proposed a plausible candidate of the general-purpose iO exploiting a cryptographic multilinear map. This construction consists of three steps; transforms a circuit into a (matrix) branching program (BP), randomize a branching program while preserving functionalities to blow-up the security, and encode an randomized branching program using a cryptographic multilinear map. This first candidate of iO has ignited the various subsequent studies [BR14, PST14, AGIS14, BGK+14, MSW14, Zim15, AB15, BMSZ16, GMM+16, DGG+18, CVW18, BGMZ18] by changing steps of a transformation and a randomization processes, all of which stand on the cryptographic multilinear maps.

To date, there are three plausible candidates of multilinear map; the first is due to Garg, Gentry, and Halevi [GGH13a] (GGH13), the second is due to Coron, Lepoint, and Tibouchi [CLT13] and the last is due to Gentry, Gorbunov, and Halevi [GGH15]. These constructions are not known to have the desired security of the multilinear map due to the specialized attack, typed zeroizing attacks [CHL+15, HJ16, CLLT16]; these attacks commonly use several encodings of zero to show the insecurity of the multi-party key exchange protocol instantiated by candidates of the multilinear map.

However, zeroizing attacks do not damage the security of current iO constructions from the candidate multilinear maps since all iO candidates do not publish 'low-level encodings of zero' which are key ingredi-

ents to break cryptographic multilinear maps. On the other hand, some iO candidates [BR14, BGK+14, AGIS14, Zim15, MSW14] claimed the provable security under the idealized multilinear map model, so-called the *generic multilinear map model*. In addition, some works have been tried to overcome this gap between idealized model and concrete instantiation of multilinear maps by presenting a concept of weak multilinear map [GMM+16, MZ18, BGMZ18].

Despite the provable security under these models, the security of concrete instantiation of indistinguishability obfuscations based on GGH13, CLT13 and GGH15 is still in dubious nature. Indeed, there have been numerous attacks to indistinguishability obfuscations which employ relations between the top level encodings of zero [CGH+15, MSZ16, ADGM17, CGH17, CLLT17, Pel18, CHKL18a, CHKL18b, CVW18, KL19, CCH+19].

However, the security of a few branching programs iO still remains as an open problem. For example, CVW and BGMZ obfuscations proposed by Chen *et al.* [CVW18] and Bartusek *et al.* [BGMZ18], which are branching program iO based on GGH15, are robust against all known (quantum) attacks. Moreover, the security of FRS obfuscation proposed by Fernando *et al.* [FRS17] when it is instantiated by CLT13 is still open. In case of branching program iO over GGH13, the GGHRSW iO [GGH+13b], the first candidate, and the GMMSSZ iO [GMM+16], a provably secure under weak GGH13 multilinear map model, are standing against all known classical attacks.

1.2 Contributions

In this thesis, we propose new polynomial cryptanalyses of branching program obfuscations based on cryptographic multilinear maps, GGH13 and GGH15.

1.2.1 Mathematical Analysis of iO based on GGH13

We present distinguishing attacks on candidates BP iO over GGH13 multilinear map based on the algorithm to solve the NTRU problem. With the novel two techniques, *program converting* and *matrix zeroizing attack*, we show that existing general-purpose BP obfuscations cannot achieve the desired security when the obfuscations use GGH13 with proposed parameters in [GGH13a, LSS14, ACLL15]. In other words, there are two functionally equivalent BPs with same length such that their obfuscations obtained by an existing BP obfuscations over GGH13 can be distinguished in polynomial time for the suggested parameters.

Our attack is applicable to wide range of obfuscations and BPs compared to the previous attacks. In particular, we show that multi-input BP obfuscations including GMMSSZ construction are insecure in the NTRU-solvable parameter regime. Further, we show that the first candidate indistinguishability obfuscation GGHRSW based on GGH13 with current parameters also does not have the desired security even if it only obfuscates input-unpartitionable BPs including branching programs generated by Barrington's theorem. Although a new property of BPs called *linear relationally inequivalence* is exploited in our attack, we show that various pairs of BPs satisfy this property.

As a result, we show that the BP obfuscations based on GGH13 mul-

tilinear map with suggested parameters are broken using the algorithm for NTRU solely. Therefore the underlying lattice dimension n of GGH13 should be set to $n = \tilde{\Theta}(\kappa^2\lambda)$ to maintain 2^λ security of obfuscation schemes. This implies the iO based on GGH13 is even much inefficient than the previous results [LMA+16, ABD16].

1.2.2 Mathematical Analysis of iO based on GGH15

We give a new polynomial time cryptanalysis, *statistical zeroizing attack*, on the candidates of iO based on the GGH15 multilinear map. This attack directly distinguishes the distributions from zeros of obfuscated programs instead of finding algebraic relations of evaluations. We particularly exploit the sample variance as a distinguisher of the distributions, while this attack introduces wide class of distinguishing methods. In particular, under an assumption on lattice preimage sampling algorithm with a trapdoor, our attack breaks the security of

- CVW obfuscation for the optimal parameter choice. Further, our attack still works for the relatively small variance σ^2 of Gaussian distribution such as $\sigma = \mathsf{poly}(\lambda)$ for the security parameter λ, and
- BGMZ obfuscation for large variance of Gaussian distribution, e.g. $\sigma = 2^\lambda$, which still enables the security proof in the weak GGH15 multilinear map model.*

This result refutes the open problem posed in [CVW18] in a certain parameter regime: the CVW obfuscation is not secure even when the adversary gets oracle access to the honest evaluations as matrix products instead of obfuscated program.

*That is, our attack is lying outside the considered attack class in [BGMZ18].

Our attack leads a new perspective to the study of iO: we should focus on the statistical properties such as shapes of distributions as well to achieve indistinguishability obfuscation. In particular, the distributions of evaluations should be (almost) the same regardless of the choice of target branching program. Previously, most attacks and constructions only focused on the algebraic structure of evaluations.

1.3 List of Papers

This thesis contains the results of the following papers.

- [CHKL18a] Jung Hee Cheon, Minki Hhan, Jiseung Kim, Changmin Lee. Cryptanalyses of Branching Program Obfuscations over GGH13 Multilinear Map from the NTRU Problem. In Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III, pages 184–210, 2018.

- [CCH$^+$19] Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, Statistical Zeroizing Attack: Cryptanalysis of Candidates of BP Obfuscation over GGH15 Multilinear Map. In Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III, pages 253–283, 2019

Chapter 2

Preliminaries

In this chapter, we introduce some information related to the thesis. In particular, we recall a concept of cryptographic multilinear map, branching program and indistinguishability obfuscation commonly used in the thesis.

2.1 Basic Notations

Throughout this thesis, let $\mathbb{N}, \mathbb{Z}$ and $\mathbb{R}$, respectively, be sets of natural numbers, integers, and real numbers.

Lower bold letters usually indicate row vectors or ring elements, and capital bold letters denote matrices. In addition, capital italic letters denote random matrices or random variables. The notation $(\mathbf{a}||\mathbf{b})$ means a concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$. The disjoint union and intersection of two sets X and Y are denoted by respectively, $X \bigsqcup Y$ and $X \bigcap Y$.

For a vector $\mathbf{v}$, the ℓ_p norm of a vector $\mathbf{v} = (v_i)$ is denoted by $\|\mathbf{v}\|_p = (\sum_i |v_i|^p)^{1/p}$. Similarly, we let $\|\mathbf{A}\|_\infty$ be the infinity norm of a matrix $\mathbf{A}$, $\|\mathbf{A}\|_\infty = \max_{i,j} a_{i,j}$ with $\mathbf{A} = (a_{i,j})$. Similarly, we can define a size of

polynomial ring element as a ℓ_2 norm of the coefficient vector.

For sampling algorithms, we usually use the 'left-arrow' notation. A notation $\mathbf{x} \leftarrow \chi$ indicates denote the operation of sampling element $\mathbf{x}$ from the distribution χ. In particular, if χ is the uniform distribution on a finite set $\mathbf{X}$, we denote $\mathbf{x} \leftarrow U(\mathbf{X})$.

2.2 Indistinguishability Obfuscation

We review the formal definition of indistinguishability obfuscation (iO).

Definition 2.2.1 (Indistinguishability Obfuscation)**.** *A probabilistic polynomial time machine $\mathcal{O}$ is an indistinguishability obfuscation for a circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:*

- *For all security parameters $\lambda \in \mathbb{N}$, for all circuits $C \in \mathcal{C}_\lambda$, for all inputs $\mathbf{x}$, the following probability holds:*

$$\Pr\left[C'(\mathbf{x}) = C(\mathbf{x}) : C' \leftarrow \mathcal{O}(\lambda, C)\right] = 1.$$

- *For any p.p.t distinguisher D, there exists a negligible function α satisfying the following statement: For all security parameters $\lambda \in \mathbb{N}$ and all pairs of circuits C_0, $C_1 \in \mathcal{C}_\lambda$, $C_0(\mathbf{x}) = C_1(\mathbf{x})$ for all inputs $\mathbf{x}$ implies*

$$\left|\Pr\left[D(\mathcal{O}(\lambda, C_0)) = 1\right] - \Pr\left[D(\mathcal{O}(\lambda, C_1)) = 1\right]\right| \leqslant \alpha(\lambda).$$

2.3 Cryptographic Multilinear Map

Boneh and Silverberg [BS03] proposed a concept which is a natural generalization of cryptographic bilinear map*, named cryptographic multilinear map. The new primitive implies numerous applications such as a multi party key exchange and a broadcast encryption. We first recall its formal definition

Definition 2.3.1 (Cryptographic Multilinear Map)**.** *Let $G_1, \cdots, G_\kappa$ and G_T be multiplicative groups of the same same order. A cryptographic κ-multilinear map is function $e : G_1 \times G_2 \times \cdots \times G_\kappa \to G_T$ such that*

1. *For any $a_1, \cdots, a_\kappa \in \mathbb{Z}$ and $(g_1, \cdots, g_\kappa) \in G_1 \times \cdots \times G_\kappa$, we have*

$$e(g_1^{a_1}, \cdots, g_\kappa^{a_\kappa}) = e(g_1, \cdots, g_\kappa)^{\prod_{i=1}^{\kappa} a_i}$$

2. *If g_i is a generator of a group G_i for each $i \in [\kappa]$, then $e(g_1, \cdots, g_\kappa)$ is also a generator of a group G_T.*

Moreover, for such groups G_i's, a discrete logarithm problem must be hard because of the security issue.

However, constructing a secure cryptographic multilinear map with $\kappa > 2$ has been a challenge problem. There exist only three main candidates called GGH13, CLT13 and GGH15, respectively [GGH13a, CLT13, GGH15], but their security is still unclear. Actually, such candidates have different structures, called graded encoding systems which is slight generalizations of a cryptographic multilinear maps. However, in this thesis, we will regard these candidates as multilinear maps.

*Cryptographic 2-multilinear map

The three main candidates are based on different structures: GGH13 is based on ideals of polynomial rings, CLT13 is based on integers, and GGH15 is based on graphs, respectively. We will defer descriptions of these candidates in the each chapter.

2.4 Matrix Branching Program

A matrix branching program (BP) is the set which consists of an index-to-input function and several matrix chains.

Definition 2.4.1. *A width w, length h, and a s-ary matrix branching program $\mathbf{P}$ over a ℓ-bit input is a set which consists of index-to-input maps $\{\mathsf{inp}_\mu : [h] \to [\ell]\}_{\mu\in[s]}$, sequences of matrices, and two disjoint sets of target matrices*

$$\mathbf{P} = \{(\mathsf{inp}_\mu)_{\mu\in[s]}, \{\mathbf{P}_{i,\mathbf{b}} \in \{0,1\}^{w\times w}\}_{i\in[h],\mathbf{b}\in\{0,1\}^s}, \mathcal{P}_0, \mathcal{P}_1 \subset \mathbb{Z}^{w\times w}\}.$$

The evaluation of $\mathbf{P}$ on input $\mathbf{x} = (x_i)_{i\in[\ell]} \in \{0,1\}^\ell$ is computed by

$$\mathbf{P}(\mathbf{x}) = \begin{cases} 0 & \text{if } \prod_{i=1}^{h} \mathbf{P}_{i,(x_{\mathsf{inp}_\mu(i)})_{\mu\in[s]}} \in \mathcal{P}_0 \\ 1 & \text{if } \prod_{i=1}^{h} \mathbf{P}_{i,(x_{\mathsf{inp}_\mu(i)})_{\mu\in[s]}} \in \mathcal{P}_1 \end{cases}.$$

When $s = 1$ ($s = 2$), the BP is called a single-input (dual-input) BP. If $s \geqslant 3$, the BP is called a multi-input BP. In this paper, we usually set $\mathcal{P}_0 = \mathbf{0}^{w\times w}$ or $\mathbf{I}$ and $\mathcal{P}_1 = \mathbb{Z}^{w\times w}\backslash\mathcal{P}_0$. Also, we call $\{\mathbf{P}_{i,\mathbf{b}}\}_{\mathbf{b}\in\{0,1\}^s}$ the i-th layer of the BP. Moreover, some branching programs have a additional structure, called a bookend vector, to change evaluations of branching programs into a constant or a vector. If it requires to describe obfuscations, we introduce it later. Remark that each obfuscation targeted in the thesis take as input

different BP type (e.g. single and dual BP) and the required properties of BP are slightly different. Therefore, we will mention the required properties used to construct an obfuscation again before describing each obfuscation.

2.5 Tensor product and vectorization

For any two matrices $A = (a_{ij})_{i,j} \in \mathbb{Z}^{m\times n}$ and $B \in \mathbb{Z}^{p\times q}$, a tensor product of matrices $A \otimes B$ is defined as a $mp \times nq$ integer matrix such that

$$A \otimes B := \begin{pmatrix} a_{11} \cdot B & \cdots & a_{1m} \cdot B \\ \vdots & \ddots & \vdots \\ a_{n1} \cdot B, & \cdots, & a_{nm} \cdot B \end{pmatrix}.$$

Consider a matrix $C \in \mathbb{Z}^{n\times m}$ whose i-th column is denoted by $\mathbf{c}_i$. Then, $\mathsf{vec}(C)$ is a mn-dimensional vector such that

$$\mathsf{vec}(C) = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_m \end{pmatrix} \in \mathbb{Z}^{mn}.$$

Then, for appropriate matrices A, B and C, the identity holds [Lau05, CLLT17] that

$$\mathsf{vec}(A \cdot B \cdot C) = (C^T \otimes A) \cdot \mathsf{vec}(B).$$

Throughout this paper, we call it 'the vectorization identity'.

2.6 Background Lattices

A lattice $\mathcal{L}$ of dimension n is a discrete additive subgroup of $\mathbb{R}^n$. If $\mathcal{L}$ is generated by the set $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$, all elements in $\mathcal{L}$ are of the form $\sum_{i=1}^{n} x_i \cdot \mathbf{b}_i$ for some integers x_i's. In this case, the lattice $\mathcal{L}$ is called the full rank lattice. Now we give several definitions and lemmas used in this paper.

For any $\sigma > 0$, the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter σ is defined as

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}-\mathbf{c}\|/\sigma^2} \text{ for all } \mathbf{x} \in \mathbb{R}^n.$$

Definition 2.6.1 (Discrete Gaussian Distribution on Lattices)**.** *For any element $\mathbf{c} \in \mathcal{R}^n$, $\sigma > 0$ and any full rank lattice $\mathcal{L}$ of $\mathcal{R}^n$, the discrete Gaussian distribution over $\mathcal{L}$ is defined as*

$$D_{\mathcal{L},\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\mathcal{L})} \text{ for all } \mathbf{x} \in \mathcal{L}$$

where $\rho_{\sigma,\mathbf{c}}(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$.

Lemma 2.6.1 ([MP12])**.** *For integers $n \geqslant 1$, $q \geqslant 2$ and $m \geqslant 2n\log q$, there is a p.p.t algorithm $\mathsf{TrapSam}(1^n, 1^m, q)$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor τ such that $\mathbf{A}$ is statistically indistinguishable from $U(\mathbb{Z}_q^{n\times m})$ with a trapdoor τ.*

Lemma 2.6.2 ([GPV08])**.** *There is a p.p.t. algorithm $\mathsf{Sample}(\mathbf{A}, \tau, \mathbf{y}, \sigma)$ that outputs a vector $\mathbf{d}$ from a distribution $D_{\mathbb{Z}^m,\sigma}$. Moreover, if $\sigma \geqslant 2\sqrt{n\log q}$, then with all but negligible probability, we have*

$$\{\mathbf{A}, \mathbf{d}, \mathbf{y} : \mathbf{y} \leftarrow U(\mathbb{Z}_q^n), \mathbf{d} \leftarrow \mathsf{Sample}(\mathbf{A}, \tau, \mathbf{y}, \sigma)\} \approx_s \{\mathbf{A}, \mathbf{d}, \mathbf{y} : \mathbf{d} \leftarrow D_{\mathbb{Z}^m,\sigma}, \mathbf{Ad} = \mathbf{y}\}.$$

Chapter 3

Mathematical Analysis of Indistinguishability Obfuscation based on the GGH13 Multilinear Map

In this chapter, we propose cryptanalyses of all existing indistinguishability obfuscation candidates based on branching programs over GGH13 multilinear map for all recommended parameter settings.

To achieve this, we introduce two novel techniques, *program converting* using NTRU-solver and *matrix zeroizing*, which can be applied to a wide range of obfuscation constructions and BPs compared to previous attacks. We then prove that, for the suggested parameters, the existing general-purpose BP obfuscations over GGH13 do not have the desired security. Especially, the first candidate indistinguishability obfuscation with input-unpartitionable branching programs (FOCS'13) and the recent BP

obfuscation (TCC'16) are not secure against our attack when they use the GGH13 with recommended parameters. Previously, there has been no known polynomial time attack for these cases.

Our attack shows that the lattice dimension of GGH13 must be set much larger than previous thought in order to maintain security. More precisely, the underlying lattice dimension of GGH13 should be set to $n = \tilde{\Theta}(\kappa^2\lambda)$ to rule out attacks from the subfield algorithm for NTRU where κ is the multilinearity level and λ the security parameter.

3.1 Preliminaries

3.1.1 Notations

Throughout this chapter, we use the bold letters to denote matrices, vectors and elements of ring. For $\mathbf{a} = a_0 + \cdots + a_{n-1} \cdot X^{n-1} \in \mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1\rangle$, where n is a power of 2, the size of $\mathbf{a}$ means the Euclidean norm of the coefficient vector $(a_0, \cdots, a_{n-1})$. We denote (j, k)-th entry of matrix $\mathbf{M}$ by $\mathbf{M}[j, k]$.

3.1.2 GGH13 Multilinear Map

Garg *et al.* suggested a candidate of multilinear map over ideal lattice [GGH13a] which is used to realize the first plausible candidate of indistinguishable obfuscation [GGH+13b]. In this section, we briefly describe the GGH13 multilinear map. For more details, we recommend readers to refer the original paper [GGH13a]. Any parameters of multilinear maps are induced by the multilinearity parameter κ and the security parameters λ. For the sake of simplicity, we denote the multilinear maps which has the previous men-

tioned parameter as (κ, λ)-GGH multilinear map.

The multilinear map is sometimes called the graded encoding scheme. *i.e.*, All encodings of message have corresponding levels. Let $\mathbf{g}$ be a secret element in $\mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1\rangle$ and q a large integer. Then, the message space and encoding space are set by $\mathcal{M} = \mathcal{R}/\langle \mathbf{g}\rangle$ and $\mathcal{R}_q = \mathcal{R}/\langle q\rangle$, respectively. In order to represent a level of encodings, the set of secret invertible elements $\mathbb{L} = \{\mathbf{z}_i\}_{1\leqslant i\leqslant \kappa} \subset \mathcal{R}_q$ is chosen. We call a subset of $\mathbb{L}$ *level set* and elements in $\mathbb{L}$ *level parameters*.

For a small message $\mathbf{m} \in \mathcal{M}$, level-$L(\subset \mathbb{L})$ encoding of $\mathbf{m}$ is:

$$\mathsf{enc}_L(\mathbf{m}) = \left[\frac{\mathbf{r}\cdot\mathbf{g} + \mathbf{m}}{\prod_{i\in L}\mathbf{z}_i}\right]_q,$$

where $\mathbf{r} \in \mathcal{R}$ is a small random element. We call $\mathsf{enc}_{\mathbb{L}}(\mathbf{m})$, $\mathsf{enc}_{\{\mathbf{z}_i\}}(\mathbf{m})$ a top-level and level 1 encoding of $\mathbf{m}$, respectively. In addition, for a matrix $\mathbf{M}$, we denote a matrix whose entries are level-L encodings of corresponding entries of $\mathbf{M}$ by $\mathsf{enc}_L(\mathbf{M})$.

The arithmetic operations between encodings are defined as follows:

$$\begin{aligned}\mathsf{enc}_L(\mathbf{m}_1) + \mathsf{enc}_L(\mathbf{m}_2) &= \mathsf{enc}_L(\mathbf{m}_1 + \mathbf{m}_2),\\ \mathsf{enc}_{L_1}(\mathbf{m}_1)\cdot\mathsf{enc}_{L_2}(\mathbf{m}_2) &= \mathsf{enc}_{L_1\sqcup L_2}(\mathbf{m}_1\cdot\mathbf{m}_2).\end{aligned}$$

Additionally, the (κ, λ)-GGH scheme provides a zerotesting parameter which can be used to determine whether a hidden message of a top-level encoding is zero or not. The zerotesting parameter $\mathbf{p}_{zt}$ is of the form:

$$\mathbf{p}_{zt} = \left[\mathbf{h}\cdot\frac{\prod_{i\in\mathbb{L}}\mathbf{z}_i}{\mathbf{g}}\right]_q,$$

where $\mathbf{h}$ is an $O(\sqrt{q})$-size element of $\mathcal{R}$. Given a top-level encoding of zero $\mathsf{enc}_{\mathbb{L}}(\mathbf{0}) = [\mathbf{r} \cdot \mathbf{g}/\prod_{i\in\mathbb{L}} \mathbf{z}_i]_q$, a zerotesting value is:

$$[\mathbf{p}_{zt} \cdot \mathsf{enc}_{\mathbb{L}}(\mathbf{0})]_q \;=\; \left[\mathbf{h} \cdot \frac{\prod_{i\in\mathbb{L}} \mathbf{z}_i}{\mathbf{g}} \cdot \frac{\mathbf{r}\cdot\mathbf{g}}{\prod_{i\in\mathbb{L}} \mathbf{z}_i}\right]_q = [\mathbf{h}\cdot\mathbf{r}]_q = \mathbf{h}\cdot\mathbf{r} \in \mathcal{R}.$$

We remark that a zerotesting value for a top-level encoding of nonzero gives an element of the form $[\mathbf{h} \cdot (\mathbf{r} + \mathbf{m} \cdot \mathbf{g}^{-1})]_q$, which is not small by Lemma 4 in [GGH13a]. Thus one can decide whether a message is zero or not by the zerotesting value.

Several papers [GGH13a, LSS14, ACLL15] proposed the parameters of (κ, λ)-GGH13 multilinear map. Here we introduce the minimum conditions that satisfy the three works.

- $\log q = \tilde{\Theta}(\kappa \cdot \log n)$
- $n = \tilde{\Theta}(\kappa^{\epsilon} \cdot \lambda^{\delta})$ for constants δ, ϵ
- $M = \tilde{O}(n^{\Theta(1)})$

Here M is the size bound of numerators $\mathbf{r}\cdot\mathbf{g} + \mathbf{m}$ of level 1 encodings.* We note that the suggested parameters in [LSS14, ACLL15] choose $\delta = \epsilon = 1$, which enables the subexponential attack with respect to λ for small κ [ABD16, BEF+17]. When $\delta \geqslant 2$, all known direct attacks on GGH13 multilinear map require exponential time for classical adversary.

*The coefficients of random values are usually sampled from the Gaussian distribution. This do not hurt the result of this paper because the coefficients are bounded with overwhelming probability.

3.2 Main Theorem

In this section, we present the results from our attacks. We denote the obfuscation within our attack range as *the attackable obfuscation*, which is formally defined by *the attackable model* in the next section. The attackable obfuscation model encompasses all suggested BP obfuscations based on GGH13 multilinear map.

Proposition 3.2.1 (Universality of the Attackable Model)**.** *BP obfuscations [GGH+13b, AGIS14, BGK+14, PST14, MSW14, GMM+16, BMSZ16] satisfy all the constraints of the attackable model.*[†]

As a result, we obtain the following main theorem.

Theorem 3.2.1. *Let $\mathcal{O}$ be an attackable obfuscator, κ, λ be the multilinearity level and the security parameter of underlying GGH13 multilinear map. Suppose that the modulus q, dimension n, size bound M of numerators of level 1 encoding of underlying GGH13 satisfy $\log q = \tilde{\Theta}(\kappa \cdot \log n), M = \tilde{O}(n^{\Theta(1)})$. Then the following propositions hold:*

1. *For $n = \tilde{\Theta}(\kappa \cdot \lambda^{\delta})$ for a constant δ as in [GGH13a, LSS14, ACLL15], there exist two functionally equivalent branching programs with $\Omega(\lambda^{\delta})$-length such that their obfuscated programs by $\mathcal{O}$ can be distinguished with high probability in polynomial time with respect to λ.*

2. *Moreover, for new parameter constraints $n = \tilde{\Theta}(\kappa^{\epsilon} \cdot \lambda^{\delta})$ for constants $\epsilon < 2, \delta$, there exist two functionally equivalent branching programs with $\Omega(\lambda^{\delta/(2-\epsilon)})$-length such that their obfuscated programs by $\mathcal{O}$ can*

[†]We deal with easier model in the main body for simplicity. We can extend the model to capture the construction in [BR14]. This extended model is placed in Appendix 6.1.1.

be distinguished with high probability in polynomial time with respect to λ.

The main theorem is proven by combining *converting program technique* and *matrix zeroizing attack* which are described in Section 3.4, 3.5. The bottleneck of the attack is the algorithm for NTRU, which is exploited in the middle step of converting technique; the other process can be done in polynomial time, while the time complexity to solve the NTRU problem relies on the parameters. The detailed analysis for the time complexity will be discussed in Section 3.4.3.

3.3 Attackable BP Obfuscations

In this section, we present a new BP obfuscation model which is attackable by our attack, *the attackable model*. We call a BP obfuscation captured by our model an *attackable BP obfuscation*.

The attackable model is composed of two steps; for a given BP, randomize BP, and encode randomized BPs by GGH13 multilinear map. More precisely, for a given branching program BP of the form

$$P = \left\{\mathbf{M}_{i,\mathbf{b}} \in \mathbb{Z}^{d_i \times d_{i+1}}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w},$$

we randomize P by several methods satisfying Definition 3.3.1 which will be described later. And then we encode each entries of randomized matrices

and outputs the obfuscated program as the set

$$\begin{aligned}\mathcal{O}(P) &= \left\{\widetilde{\mathbf{S}}, \widetilde{\mathbf{S}}' \in \mathcal{R}_q^{d_0 \times (d_1+e_1)}\right\} \\ &\cup \left\{\{\widetilde{\mathbf{M}}_{i,\mathbf{b}}, \widetilde{\mathbf{M}}'_{i,\mathbf{b}} \in \mathcal{R}_q^{(d_i+e_i)\times(d_{i+1}+e_{i+1})}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}, \right\} \\ &\cup \left\{\widetilde{\mathbf{T}}, \widetilde{\mathbf{T}}' \in \mathcal{R}_q^{(d_{\ell+1}+e_{\ell+1})\times d_{\ell+2}}\right\}\end{aligned}$$

and the public parameters of GGH13 multilinear map. $\mathbf{S}, \mathbf{T}$ denote bookend matrices, and matrices with apostrophe mean the matrices of dummy program. In the attackable model, we specify the following property instead of establishing how to evaluate the program exactly. To evaluate the input value, a new function $Eval_{\widetilde{\mathbf{M}}} : \{0,1\}^N \to \mathcal{R}_q^{d_0 \times d_{\ell+2}}$ is computed as follows:

$$Eval_{\widetilde{\mathbf{M}}}(\mathbf{x}) = \widetilde{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \widetilde{\mathbf{M}}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \widetilde{\mathbf{T}} - \widetilde{\mathbf{S}}' \cdot \prod_{i=1}^{\ell} \widetilde{\mathbf{M}}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \widetilde{\mathbf{T}}' \in \mathcal{R}_q^{d_0 \times d_{\ell+2}}.$$

Proposition 3.3.1 (Evaluation of Obfuscation)**.** *For a program P and program $\mathcal{O}(P)$ obfuscated by the attackable model, the evaluation of $\mathcal{O}(P)$ at a root $\mathbf{x}$ of P yields a top-level GGH13 encoding of zero in specific entry of the matrix $Eval_{\widetilde{\mathbf{M}}}(\mathbf{x})$. In other words, there are two integers u, v such that $Eval_{\widetilde{\mathbf{M}}}(\mathbf{x})[u, v]$ is an encoding of zero at level $\mathbb{L}$ for every input $\mathbf{x}$ satisfying $P(\mathbf{x}) = 0$.*

In the rest of this section, we explain specified descriptions of the attackable model in Section 4.1 and 4.2, and present a constraint of BPs to execute our attack in Section 4.3.

3.3.1 Randomization for Attackable Obfuscation Model

We introduce the conditions for BP randomization of attackable obfuscation model. These conditions for randomization covers all of the BP randomization methods suggested in the first candidate *iO* [GGH+13b] and its subsequent works [AGIS14, BGK+14, PST14, MSW14, GMM+16, BMSZ16]. In other words, higher dimension embedding, scalar bundling, Kilian randomization, bookend matrices (vectors), and dummy programs are captured by the attackable conditions.

Definition 3.3.1 (Attackable Conditions for Randomization)**.** *For a branching program* $P = \left\{\mathbf{M}_{i,\mathbf{b}} \in \mathbb{Z}^{d_i \times d_{i+1}}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}$*, the attackable randomized branching program is the set*

$$\begin{aligned} \mathit{Rand}(P) &= \left\{\mathbf{R}_{\mathbf{S}}, \mathbf{R}'_{\mathbf{S}} \in \mathbb{Z}^{d_0\times(d_1+e_1)}\right\} \\ &\cup \left\{\left\{\mathbf{R}_{i,\mathbf{b}}, \mathbf{R}'_{i,\mathbf{b}} \in \mathbb{Z}^{(d_i+e_i)\times(d_{i+1}+e_{i+1})}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}, \right\} \\ &\cup \left\{\mathbf{R}_{\mathbf{T}}, \mathbf{R}'_{\mathbf{T}} \in \mathbb{Z}^{(d_{\ell+1}+e_{\ell+1})\times d_{\ell+2}}\right\} \end{aligned}$$

satisfying the following properties, where $d_0, d_{\ell+2}, e_i$*'s are integers.*
1. There exist matrices $\mathbf{S}_0, \mathbf{S}'_0 \in \mathbb{Z}^{d_0\times d_1}, \mathbf{T}_0, \mathbf{T}'_0 \in \mathbb{Z}^{d_\ell \times d_{\ell+1}}$ *and scalars* $\alpha_{\mathbf{S}}, \alpha'_{\mathbf{S}}, \alpha_{\mathbf{T}}, \alpha'_{\mathbf{T}}, \{\alpha_{i,\mathbf{b}}, \alpha'_{i,\mathbf{b}}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}$ *such that the following equations hold for all* $\{\mathbf{b}_i \in \{0,1\}^w\}_{i\in[\ell]}$*:*

$$\mathbf{R}_S \cdot \prod_{i=1}^{\ell} \mathbf{R}_{i,\mathbf{b}_i} \cdot \mathbf{R}_T = \alpha_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha_{i,\mathbf{b}_i} \cdot \alpha_{\mathbf{T}} \cdot \left(\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0\right),$$
$$\mathbf{R}'_S \cdot \prod_{i=1}^{\ell} \mathbf{R}'_{i,\mathbf{b}_i} \cdot \mathbf{R}'_T = \alpha'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha'_{i,\mathbf{b}_i} \cdot \alpha'_{\mathbf{T}} \cdot \left(\mathbf{S}'_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}'_{i,\mathbf{b}_i} \cdot \mathbf{T}'_0\right).$$

2. The evaluation of randomized program is done by checking whether the

fixed entries of $RP(\mathbf{x}) := \mathbf{R_S} \cdot \prod_{i=1}^{\ell} \mathbf{R}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{R_T} - \mathbf{R'_S} \cdot \prod_{i=1}^{\ell} \mathbf{R}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{R'_T}$ *are zero or not. Especially, there are two integers* u, v *such that* $P(\mathbf{x}) = 0 \Rightarrow RP(\mathbf{x})[u, v] = 0$.

Matrices with apostrophe are called *dummy matrices*, $\mathbf{R_S}, \mathbf{R'_S}, \mathbf{R_T}, \mathbf{R'_T}$ bookend matrices (vectors), and α's *bundling scalars*. When some elements of $Rand(P)$ (or bundling scalars) are trivial elements, we say that there is no such element.

3.3.2 Encoding by Multilinear Map

After the randomization, we encode the randomized matrix branching program by GGH13 multilinear map. We stress that we *do not encode* dummy/bookend matrices if there are no dummy/bookends, respectively.

For each randomized matrices, $\mathbf{R}_{i,\mathbf{b}}, \mathbf{R}'_{i,\mathbf{b}}$ and randomized bookend matrices $\mathbf{R_S}, \mathbf{R'_S}, \mathbf{R_T}, \mathbf{R'_T}$, we obtain the encoded matrices $\mathsf{enc}_{L_{i,\mathbf{b}}}(\mathbf{R}_{i,\mathbf{b}})$ whose entries are encoding of corresponding entries of randomized matrix $\mathbf{R}_{i,\mathbf{b}}$. For brevity we write $\widetilde{\mathbf{M}}_{i,\mathbf{b}}$ to denote $\mathsf{enc}_{L_{i,\mathbf{b}}}(\mathbf{R}_{i,\mathbf{b}})$, and the other matrices $\widetilde{\mathbf{M}}'_{i,\mathbf{b}}$, $\widetilde{\mathbf{S}}$, $\widetilde{\mathbf{S}}'$, $\widetilde{\mathbf{T}}$, $\widetilde{\mathbf{T}}'$ are defined in similar manner.

Two conditions should hold in the attackable model

1. the evaluation of valid input is top-level, in other words, for all input $\mathbf{x}$, $\left(\cup_{i=1}^{\ell} L_{i,\mathbf{x}_{\mathsf{inp}(i)}}\right) \cup L_{\mathbf{S}} \cup L_{\mathbf{T}} = \mathbb{L}$ where $\mathbb{L}$ denotes top-level set,
2. the sizes of set L's are all similar, that is, there is a constant C such that $|L_{i,\mathbf{b}}|/|L_{j,\mathbf{b}'}| \leqslant C$ for all $i, j, \mathbf{b}, \mathbf{b}'$ and similar inequalities hold for $L_{\mathbf{S}}, L_{\mathbf{T}}$.

In practice, the level L's is determined by *the straddling set system* introduced in [BGK+14, MSW14], and these constructions satisfy our conditions. Using the condition 1 and Definition 3.3.1, Proposition 3.3.1 can be

easily verified. We also note that the condition 2 implies $\ell = \Theta(\kappa)$, where κ is the level of underlying multilinear map.

3.3.3 Linear Relationally Inequivalent Branching Programs

At last, we explain the condition, *linear relationally inequivalence*, for branching programs of attackable BP obfuscation. This condition is used at the last section, but we note that there are several linear relationally inequivalence BPs as stated in Proposition 3.3.2.

To define the linear relationally inequivalence, we consider evaluations of invalid inputs of branching program and denote $\prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i}$ by $\mathbf{M}(\mathbf{b})$ for $\mathbf{b} = (\mathbf{b}_1, \cdots, \mathbf{b}_\ell)$. We define linear relations of two BPs and the *linear relationally inequivalence* of BPs as

Definition 3.3.2 (Linear Relations of Branching Program)**.** *For a given branching program*

$$P_{\mathbf{M}} = \left\{\mathbf{M}_{i,\mathbf{b}} \in \mathbb{Z}^{d_i \times d_{i+1}}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w},$$

the set of linear relations of $P_{\mathbf{M}}$ is

$$L_{\mathbf{M}} := \left\{ (q_{\mathbf{b}})_{\mathbf{b}\in\{0,1\}^{w\times\ell}} : \sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} q_{\mathbf{b}} \cdot \mathbf{M}(\mathbf{b}) = \mathbf{0}^{d_1\times d_{\ell+1}} \right\}$$

Definition 3.3.3 (Linear Relationally Inequivalence)**.** *We say that two branching programs $P_{\mathbf{M}}$ and $P_{\mathbf{N}}$ with the same length are linear relationally inequivalent if $L_{\mathbf{M}} \neq L_{\mathbf{N}}$.*

The set of linear relations of a given BP is easily computed by comput-

ing the kernel, considering BP matrices as vectors. It is clear that $L_{\mathbf{M}}$ is a lattice. We note that the set of linear relations of BP is not determined by the functionality of BP, and indeed it seems that they are irrelevant.

Further, one can observe that if $P_{\mathbf{M}}, P_{\mathbf{N}}$ are linear relationally inequivalent BPs, then so do two extended BPs $P'_{\mathbf{M}}, P'_{\mathbf{N}}$ which are obtained by concatenating some other (functionally equivalent) BPs on the right (or left) of $P_{\mathbf{M}}, P_{\mathbf{N}}$. Therefore we can show that there exist arbitrary large two functionally equivalent BPs which are linear relationally inequivalent.

We conclude this section by presenting a proposition that shows concrete examples of linear relationally inequivalent BPs, which are placed in Appendix 6.1.3.

Proposition 3.3.2. *There are two functionally equivalent, but linear relationally inequivalent branching programs. Especially, there are examples satisfying the linear relationally inequivalence which are*
1) generated by Barrington's theorem and input-unpartitionable or
2) from non-deterministic finite automata and read-once, in other words, inp *is a bijection.*

3.4 Program Converting Technique

In this section, we describe the program converting technique, which remove the hindrance of modulus q and $\mathbf{g}$. We first define new notion $\mathbf{Y}$ *program (of* P*)* if all entries of branching program matrices corresponding a program P are in a space $\mathbf{Y}$ while preserving many properties. For example, the obfuscated program $\mathcal{O}(P)$ is $\mathcal{R}_q$ program. Suppose that the obfuscated program $\mathcal{O}(P)$ of program P is given.

We will convert given obfuscated program $\mathcal{O}(P)$ into $\mathcal{R}$ and $\mathcal{R}/\langle\mathbf{g}\rangle$ pro-

gram using the algorithm to solve the NTRU problem, especially *subfield attacks* [ABD16, CJL16] which solves the problem with large modulus q.

Proposition 3.4.1 ([ABD16, CJL16, CHL17, KF17])**.** *Let q be a large integer, n a power of two, M a constant much smaller than q, $\mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1\rangle$ and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. For a given $[\mathbf{f}_1/\mathbf{f}_2]_q \in \mathcal{R}_q$ for $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{R}$ with size smaller than M, there is an algorithm to compute $(\mathbf{c}\cdot\mathbf{f}_2, \mathbf{c}\cdot\mathbf{f}_1) \in \mathcal{R}^2$ such that sizes of $\mathbf{c}$, $\mathbf{c}\cdot\mathbf{f}_1$ and $\mathbf{c}\cdot\mathbf{f}_2$ are much smaller than q in time $2^{O(\beta)}\cdot poly(n)$ for a constant β satisfying $\beta/\log\beta = \Theta(n\log M/\log^2 q)$.*

We note that the similar results hold for other non-cyclotomic ring [KF17, CHL17] or for $\mathbf{f}_1, \mathbf{f}_2$ from certain distribution [ABD16]. Throughout in this paper, we only consider the bounded coefficient $\mathbf{f}_1\mathbf{f}_2$ in cyclotomic ring for brevity.

For given obfuscated program in $\mathcal{R}_q$, we first make the NTRU instances and solve the problem, and then convert to $\mathcal{R}$ program by some computations on obfuscated matrices. This procedure replaces the level parameter $\mathbf{z}_i$ with a small element $\mathbf{c}_i$. The $\mathcal{R}$ program preserves same functionality with the $\mathcal{R}_q$ program. Subsequently, we convert this $\mathcal{R}$ program to $\mathcal{R}/\langle\mathbf{g}\rangle$ program by recovering the ideal $\langle\mathbf{g}\rangle$.

3.4.1 Converting to $\mathcal{R}$ Program

In order to remove the modulus q, we employ the algorithm for solving NTRU problem. Let $\widetilde{\mathbf{M}}_{i,\mathbf{b}}$ be the obfuscated matrix of $\mathbf{R}_{i,\mathbf{b}}$. Then, each (j,k)-th entries of obfuscated matrix $\widetilde{\mathbf{M}}_{i,\mathbf{b}}$ is of the form

$$\mathbf{d}_{j,k,\mathbf{b}} = \left[\frac{\mathbf{r}_{j,k,\mathbf{b}}\cdot\mathbf{g} + \mathbf{a}_{j,k,\mathbf{b}}}{\mathbf{z}_i}\right]_q,$$

where $\mathbf{a}_{j,k,\mathbf{b}}$ is the (j,k)-th entry of the matrix $\mathbf{R}_{i,\mathbf{b}}$ and $\mathbf{r}_{j,k,\mathbf{b}} \in \mathcal{R}$ are random small elements. Consider an element $\mathbf{v} = [\mathbf{d}_{1,1,\mathbf{0}}/\mathbf{d}_{1,2,\mathbf{0}}]_q = [(\mathbf{r}_{1,1,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{1,1,\mathbf{0}})/(\mathbf{r}_{1,2,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{1,2,\mathbf{0}})]_q$. Then, $\mathbf{v}$ is the instance of the NTRU problem since the size of denominator and numerator of $\mathbf{v}$ is much smaller than q in the parameter setup of GGH13 multilinear map.

Applying Proposition 3.4.1 to an instance $\mathbf{v}$, one can find a pair $(\mathbf{c}_i \cdot (\mathbf{r}_{1,1,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{1,1,\mathbf{0}}),\ \mathbf{c}_i \cdot (\mathbf{r}_{1,2,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{1,2,\mathbf{0}})) \in \mathcal{R}^2$ with relatively small $\mathbf{c}_i \in \mathcal{R}$. Further, for any element $\mathbf{d}_{j,k,\mathbf{b}} \in \widetilde{\mathbf{M}}_{i,\mathbf{b}}$, we can remove the modulus q by computing

$$\mathbf{c}_i \cdot (\mathbf{r}_{1,1,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{1,1}, \mathbf{0}) \cdot [\mathbf{d}_{j,k,\mathbf{b}}/\mathbf{d}_{1,1,\mathbf{0}}]_q = \mathbf{c}_i \cdot (\mathbf{r}_{j,k,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{j,k,\mathbf{0}}) \in \mathcal{R}$$

because of the small size of $\mathbf{c}_i$. Consequently, one can obtain a new matrix $\mathbf{D}_{i,\mathbf{b}}$ over $\mathcal{R}$ whose (j,k)-th entry is $\mathbf{c}_i \cdot (\mathbf{r}_{j,k,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{j,k,\mathbf{0}})$.

Similarly, a new dummy matrix $\mathbf{D}'_{i,\mathbf{b}}$ over $\mathcal{R}$ can be obtained because $\widetilde{\mathbf{M}}'_{i,\mathbf{b}}$ shares the level parameter $\mathbf{z}_i$ with $\widetilde{\mathbf{M}}_{i,\mathbf{b}}$ by multiplying $\mathbf{c}_i \cdot (\mathbf{r}_{j,k,\mathbf{0}} \cdot \mathbf{g} + \mathbf{a}_{j,k,\mathbf{0}})$ to $[\mathbf{d}'_{j,k,\mathbf{b}}/\mathbf{d}_{1,1,\mathbf{0}}]_q$ where $\mathbf{d}'_{j,k,\mathbf{b}}$ is a (j,k)-th entry of $\widetilde{\mathbf{S}}'_{i,\mathbf{b}}$. We easily observe that $2 \cdot 2^w$ matrices $\mathbf{D}_{i,\mathbf{b}}$ and $\mathbf{D}'_{i,\mathbf{b}}$ share the parameter $\mathbf{c}_i$.

For all matrices $\widetilde{\mathbf{M}}_{i,\mathbf{b}}$ and $\widetilde{\mathbf{M}}'_{i,\mathbf{b}}$ with $i \in [\ell]$ and $\mathbf{b} \in \{0,1\}^w$, we can obtain new matrices $\mathbf{D}_{i,\mathbf{b}}$ and $\mathbf{D}'_{i,\mathbf{b}}$ over $\mathcal{R}$. In the case of bookend matrices $\widetilde{\mathbf{S}}$ and $\widetilde{\mathbf{T}}$, they are converted into matrices over $\mathcal{R}$ with small constants $\mathbf{c}_\mathbf{S}$ and $\mathbf{c}_\mathbf{T}$, respectively. Note that this step runs in polynomial time if κ is large [ABD16, CJL16, CHL17, KF17]. Detailed analysis of this part is discussed in Section 3.4.3.

Therefore, we can convert $\mathcal{R}_q$-program $\mathcal{O}(P)$ into a new program, $\mathcal{R}$-*program of* P:

$$\mathcal{R}(P) = \{\mathbf{D}_\mathbf{S}, \mathbf{D}_\mathbf{T}, \mathbf{D'}_\mathbf{S}, \mathbf{D'}_\mathbf{T}, \{\mathbf{D}_{i,\mathbf{b}}, \mathbf{D'}_{i,\mathbf{b}}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}\}.$$

Note that the matrix $\mathbf{D}_{i,\mathbf{b}}$ of $\mathcal{R}(P)$ is of the form $\mathbf{c}_i \cdot \mathbf{R}_{i,\mathbf{b}} \pmod{\langle \mathbf{g} \rangle}$ in $\mathcal{R}/\langle \mathbf{g} \rangle$.

Dummy and bookend matrices satisfies similar relations. We denote $\mathbf{c}_i \cdot \alpha_{i,\mathbf{b}}$ and $\mathbf{c}_i \cdot \alpha'_{i,\mathbf{b}}$ by $\rho_{i,\mathbf{b}}$, $\rho'_{i,\mathbf{b}}$ for simplicity. The properties of Definition 3.3.1 is naturally extended to the following. The proposition 3.4.2 means an evaluation of $\mathcal{R}(P)$ preserves the functionality up to constant on the valid input $\mathbf{x}$.

Proposition 3.4.2 (Evaluation of $\mathcal{R}$ and $\mathcal{R}/\langle \mathbf{g} \rangle$ Branching Program)**.** *For a $\mathcal{R}$ program given in this section, the following propositions holds:*
1. The higher dimension embedding matrices $\mathbf{U}$'s are eliminated in the product of randomized matrix branching program, that is, there are matrices $\mathbf{S}_0, \mathbf{S}'_0 \in \mathbb{Z}^{d_0 \times d_1}, \mathbf{T}_0, \mathbf{T}'_0 \in \mathbb{Z}^{d_{\ell+1} \times d_{\ell+2}}$ such that the following equations hold for all input x:

$$\mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D}_{\mathbf{T}} = \rho_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i} \cdot \rho_{\mathbf{T}} \cdot \left(\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0 \right) \pmod{\langle \mathbf{g} \rangle},$$

$$\mathbf{D}'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}'_{i,\mathbf{b}_i} \cdot \mathbf{D}'_{\mathbf{T}} = \rho'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho'_{i,\mathbf{b}_i} \cdot \rho'_{\mathbf{T}} \cdot \left(\mathbf{S}'_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}'_{i,\mathbf{b}_i} \cdot \mathbf{T}'_0 \right) \pmod{\langle \mathbf{g} \rangle}.$$

2. The evaluation of $\mathcal{R}$ program is done by checking whether the fixed entries of $Eval_{\mathbf{D}}(\mathbf{x}) := \mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D}_{\mathbf{T}} - \mathbf{D}'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D}'_{\mathbf{T}}$ is multiple of $\mathbf{g}$ or not. Especially, there are two integers u, v such that $P(\mathbf{x}) = 0 \Rightarrow Eval_{\mathbf{D}}(\mathbf{x})[u, v] = 0 \pmod{\langle \mathbf{g} \rangle}$

3.4.2 Recovering $\langle \mathbf{g} \rangle$ and Converting to $\mathcal{R}/\langle \mathbf{g} \rangle$ Program

Next, we will compute a basis of the plaintext space $\langle \mathbf{g} \rangle$ to transform $\mathcal{R}$ program into $\mathcal{R}/\langle \mathbf{g} \rangle$-program. Unlike other attacks, we do not use the assumption 'input partitionability'. We exploits the fact that $\mathcal{R}$ program which comes from $\mathcal{R}_q$ program has the same functionality up to constant. However, existing attacks with input partitionable assumption and our cryptanalysis cannot be applied to a BP program for an 'evasive function' since it does not output multiples of $\mathbf{g}$. It consists of following two steps:

Finding a multiple of g. This step is done by computing $Eval_{\mathbf{D}}$ at the zeros of program P. We compute $Eval_{\mathbf{D}}(\mathbf{x})$ for $\mathcal{R}$ program $\mathcal{R}(P)$ at $\mathbf{x}$ satisfying $P(\mathbf{x}) = 0$. Then, Proposition 3.4.2 implies that $Eval_{\mathbf{D}}(\mathbf{x})[u, v]$ is a multiple of $\mathbf{g}$. More precisely, $Eval_{\mathbf{D}}(\mathbf{x})[u, v]$ is of the form

$$\mathbf{c_S} \cdot \mathbf{c_T} \cdot \prod_{i=1}^{\ell} \mathbf{c}_i \cdot \mathbf{a} \cdot \mathbf{g}$$

when $\mathbf{p}_{zt} \cdot Eval_{\widetilde{\mathbf{M}}}(\mathbf{x})[u, v] = \mathbf{a} \cdot \mathbf{h} \pmod{q}$ for some $\mathbf{a} \in \mathcal{R}$ such that $\|\mathbf{a} \cdot \mathbf{h}\|_2$ is less than $q^{3/4}$.

This procedure outputs the value which is not only multiple of $\mathbf{g}$ but also $\mathbf{c}_i$'s. However, we can generate several different $\mathcal{R}$ program from $\mathcal{O}(P)$ for different solutions of Proposition 3.4.1. We assume that the multiples of $\mathbf{g}$ from different $\mathcal{R}$ program are independent multiples of $\mathbf{g}$, with the randomized lattice reduction algorithm as in [GN08].

Computing Hermite Normal Form of $\langle \mathbf{g} \rangle$. For given several random

multiples $\mathbf{f}_i \cdot \mathbf{g}$ of $\mathbf{g}$, we can recover a basis of $\langle \mathbf{g} \rangle$ by computing sum of sufficiently many ideal $\langle \mathbf{f} \cdot \mathbf{g} \rangle$ represented by a lattice with basis $\{\mathbf{f} \cdot \mathbf{g}, \mathbf{f} \cdot \mathbf{g} \cdot X, \cdots, \mathbf{f} \cdot \mathbf{g} \cdot X^{n-1}\}$ or computing the Hermite Normal Form of union of their generating sets by applying the lemma [ABD16, Lemma 1].

Both computations are done in polynomial time in λ and κ, since the evaluations and computing the Hermite normal form has a polynomial time complexity. Eventually, we recover the basis of ideal lattice $\langle \mathbf{g} \rangle$ and we can efficiently compute the arithmetic computations in $\mathcal{R}/\langle \mathbf{g} \rangle$. In other words, we get a $\mathcal{R}/\langle \mathbf{g} \rangle$ program corresponding to $\mathcal{O}(P)$ (or P), whose properties are characterized by Proposition 3.4.2. For convenience, we abuse the notation; from now, $\mathcal{R}(P)$ is the $\mathcal{R}/\langle \mathbf{g} \rangle$ program and $\mathbf{D}_{\mathbf{S}}, \mathbf{D}_{\mathbf{T}}$ and $\mathbf{D}_{i,\mathbf{b}}$ for all $i \in [\ell], \mathbf{b} \in \{0,1\}^w$ are matrices over $\mathcal{R}/\langle \mathbf{g} \rangle$.

3.4.3 Analysis of the Converting Technique

We discuss the time complexity of our program converting technique. The program converting consists of converting to $\mathcal{R}$ program, evaluating of $\mathcal{R}$ program, computing a Hermite Normal Form of an ideal lattice $\langle \mathbf{g} \rangle$. The last two steps take polynomial time complexity, so the total cost is dominated by the first step. More precisely, solving the NTRU problem for each encoded matrix is the dominant part of the program converting.

To estimate the cost of solving the NTRU problem, we assume that each component of branching program is encoded by GGH13 multilinear map in level-1. The general cases are similar but a bit more complex when we assume that the size of level sets are not too different so that $\ell = \Theta(\kappa)$.

Suppose that an obfuscated branching program $\mathcal{O}(P)$ over (κ, λ)-GGH13 multilinear map is given. For constants δ, e and security parameter λ, multilinearity level κ, n, M, and $\log q$ are set to be $\tilde{\Theta}(\kappa^e \cdot \lambda^\delta)$, $n^{\Theta(1)}$, and

$\tilde{\Theta}(\kappa \cdot \log n)$, respectively. Proposition 3.4.1 implies that one can convert the program in $2^{O(\beta)} \cdot poly(\lambda, \kappa)$ time for $\frac{\beta}{\log \beta} = \Theta(\frac{n \log M}{\log^2 q}) = \tilde{\Theta}\left(\frac{\lambda^\delta}{\kappa^{2-e}}\right)$. Therefore, the program converting technique is done in polynomial time for $\kappa = \tilde{\Omega}(\lambda^{\delta/(2-e)})$. Alternatively, the program converting technique is done in polynomial time for obfuscated programs with length $\ell = \tilde{\Omega}(\lambda^{\delta/(2-e)})$.

We note that choosing large n to make the subfield attack work in exponential time rules out our attack as well. More concretely, if one chooses $n = \tilde{\Theta}(\kappa^2 \lambda)$ then the underlying NTRU problem is hard enough to block known subexponential time attacks.

3.5 Matrix Zeroizing Attack

In this section, we present a distinguishing attack on $\mathcal{R}$ programs to complete our cryptanalysis of attackable BP obfuscation model. We note that we can evaluate the $\mathcal{R}$ program at invalid inputs, or *mixed input*, since the multilinearity level which was the obstacle of mixed inputs is removed in the previous step. We recall that $\mathbf{M}(\mathbf{b})$ denotes $\prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i}$ for $\mathbf{b} = (\mathbf{b}_1, \cdots, \mathbf{b}_\ell)$ and the set of linear relations

$$L_{\mathbf{M}} = \left\{ (q_{\mathbf{b}})_{\mathbf{b} \in \{0,1\}^{w \times \ell}} : \sum_{\mathbf{b} \in \{0,1\}^{w \times \ell}} q_{\mathbf{b}} \cdot \mathbf{M}(\mathbf{b}) = \mathbf{0}^{d_1 \times d_{\ell+1}} \right\}$$

which was defined in Section 3.3.3. We also recall that the two program $\mathbf{M}$ and $\mathbf{N}$ are linear relationally inequivalent if $L_{\mathbf{M}} \neq L_{\mathbf{N}}$.

For two functionally equivalent but linear relationally inequivalent BPs $P_{\mathbf{M}}$ and $P_{\mathbf{N}}$, we will zeroize the $\mathbf{R}$ program corresponding to $P_{\mathbf{M}}$ by exploiting the linear relation, whereas $\mathbf{R}$ program corresponding to $P_{\mathbf{N}}$ would not be a zero matrix. The result of the matrix zeroizing attack is as follows.

Proposition 3.5.1 (Matrix Zeroizing Attack)**.** *For functionally equivalent but linear relationally inequivalent branching programs $P_{\mathbf{M}}, P_{\mathbf{N}}$, there is a PPT algorithm which can distinguish between two $\mathcal{R}$ programs $\mathcal{R}(P_{\mathbf{M}})$ and $\mathcal{R}(P_{\mathbf{N}})$ obtained by the method in Section 3.4 with non-negligible probability.*

Now we explain how to distinguish two $\mathcal{R}$ programs using linear relationally inequivalence. Despite the absence of multilinearity level, we still have obstacles to directly exploit linear relationally inequivalence: scalar bundlings. To explain the main idea of the attack, we assume that, for the time being, all scalar bundling are trivial in the obtained program in Section 5. We later explain how to deal the scalar bundlings.

Suppose that two BPs $P_{\mathbf{M}}, P_{\mathbf{N}}$ and an $\mathbf{R}$ program

$$\mathcal{R}(P_{\mathbf{X}}) = \{\mathbf{D}_{\mathbf{S}}, \mathbf{D}_{\mathbf{T}}, \mathbf{D}_{\mathbf{S}'}, \mathbf{D}_{\mathbf{T}'}, \{\mathbf{D}_{i,\mathbf{b}}, \mathbf{D'}_{i,\mathbf{b}}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}\}$$

are given. Our goal is to determine $\mathbf{X} = \mathbf{N}$ or $\mathbf{X} = \mathbf{M}$. We can compute a linear relation $(q_{\mathbf{b}})$ which is an element of $L_{\mathbf{M}} \backslash L_{\mathbf{N}}$ in polynomial time[‡] by computing a basis of kernel, and solve the membership problems of lattice for each vector in the basis. Then the following equation holds

$$\begin{aligned}
&\sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D}_{\mathbf{T}} \right) \quad = \sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0 \right) \\
&= \mathbf{S}_0 \cdot \sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \right) \cdot \mathbf{T}_0 \quad = \mathbf{S}_0 \cdot \mathbf{0}^{d_1\times d_{\ell+1}} \cdot \mathbf{T}_0 = \mathbf{0}^{d_0\times d_{\ell+2}} \pmod{\langle \mathbf{g} \rangle}
\end{aligned}$$

[‡]The dimension of $(q_{\mathbf{b}})_{\mathbf{b}\in\{0,1\}^{w\times\ell}}$ is $2^{w\times\ell}$, which is exponentially large. However, we can reduce this exponential part by considering a polynomial number of $\mathbf{b}$ so that there are linear relations.

when $\mathbf{X} = \mathbf{M}$ whereas this is not hold when $\mathbf{X} = \mathbf{N}$. Therefore, the matrix zeroizing attack works when the scalar bundlings are all trivial.

When the scalar bundlings are not trivial, we can do the similar computation after recovering ratios of bundling scalars. Assume that we know $\rho_{i,\mathbf{u}}/\rho_{i,\mathbf{v}}$ for every $1 \leqslant i \leqslant \ell$ and $\mathbf{u}, \mathbf{v} \in \{0,1\}^w$. Consequently, for $\mathbf{r}(\mathbf{b}) := \prod_{i\in[\ell]} \rho_{i,\mathbf{b}_i}$ where $\mathbf{b} = (\mathbf{b}_1, \cdots, \mathbf{b}_\ell)$, we can compute $\mathbf{r}(\mathbf{b})/\mathbf{r}(\mathbf{c})$ for $\mathbf{b}, \mathbf{c} \in \{0,1\}^{w\times\ell}$ by multiplying ratios of bundling scalars. Then, we can calculate

$$\begin{aligned}
&\sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \frac{\mathbf{r}(\mathbf{0})}{\mathbf{r}(\mathbf{b})} \cdot \mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D}_{\mathbf{T}} \right) \\
&= \sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \rho_{\mathbf{S}} \cdot \mathbf{r}(\mathbf{0}) \cdot \rho_{\mathbf{T}} \cdot \mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0 \right) \\
&= \rho_{\mathbf{S}} \cdot \mathbf{r}(\mathbf{0}) \cdot \rho_{\mathbf{T}} \cdot \mathbf{S}_0 \cdot \sum_{\mathbf{b}\in\{0,1\}^{w\times\ell}} \left(q_{\mathbf{b}} \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \right) \cdot \mathbf{T}_0 \pmod{\langle \mathbf{g} \rangle},
\end{aligned}$$

which is a zero matrix if and only if $\mathbf{X} = \mathbf{M}$.

Accordingly, we should remove the scalar bundlings or recover ratios of scalar bundlings to execute the matrix zeroizing attack. In the rest of this section, we show how to recover or remove (ratios of) scalar bundlings in several cases. In Section 3.5.2, we explain how to recover all ratios in general cases by complex techniques.

3.5.1 Existing BP Obfuscations

In this section, we show how to apply the matrix zeroizing attack on two remarkable obfuscations, GGHRSW and GMMSSZ. The other examples

on obfuscations [PST14, BMSZ16] are placed in Appendix 6.1.2.

GGHRSW.

As the first case, we consider the first BP obfuscation, GGHRSW, which has the identity dummy program. We note that the attack for this case works for the attackable BP obfuscations with fixed dummy program as well. For this case, a constraint on the bundling scalars $\alpha_{\mathbf{x}} = \alpha'_{\mathbf{x}}$ for every input $\mathbf{x}$ is given where $\alpha_{\mathbf{x}} = \alpha_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \alpha_{\mathbf{T}}$, $\alpha'_{\mathbf{x}} = \alpha'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \alpha'_{\mathbf{T}}$. Suppose $\mathcal{R}$ *program of* P is given by

$$\mathcal{R}(P) = \{\mathbf{D}_{\mathbf{S}}, \mathbf{D}_{\mathbf{T}}, \mathbf{D}_{\mathbf{S}'}, \mathbf{D}_{\mathbf{T}'}, \{\mathbf{D}_{i,\mathbf{b}}, \mathbf{D'}_{i,\mathbf{b}}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}\}.$$

By Proposition 3.4.2, the following equations hold

$$\mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D}_{\mathbf{T}} = \rho_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \rho_{\mathbf{T}} \cdot \left(\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{T}_0 \right) \bmod \langle \mathbf{g} \rangle,$$
$$\mathbf{D}'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D}'_{\mathbf{T}} = \rho'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \rho'_{\mathbf{T}} \cdot \left(\mathbf{S}'_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{T}'_0 \right) \bmod \langle \mathbf{g} \rangle.$$

Here we assume that each $\mathbf{M}'_{i,\mathbf{x}_{\mathsf{inp}(i)}}$ are identity matrices. Now we consider the two quantity of evaluations $Plain_{\mathbf{D}}(\mathbf{x}) := \mathbf{D}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D}_{\mathbf{T}}$ and $Dummy_{\mathbf{D}}(\mathbf{x}) := \mathbf{D'}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{D'}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{D'}_{\mathbf{T}}$.

According to the condition of scalar bundlings, $\rho_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \rho_{\mathbf{T}} = \rho'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \rho'_{\mathbf{T}}$ since the value $\mathbf{c}$'s are shared for plain and dummy program. It is possible to remove scalar bundlings by dividing $Plain_{\mathbf{D}}(\mathbf{x})$ by $Dummy_{\mathbf{D}}(\mathbf{x})$. In other words, we can get $\mathbf{d} \cdot \mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{T}_0$ for some fixed $\mathbf{d}$ from the above division. Since we know all $\mathbf{M}$'s, the matrix zeroizing attack works well for the computed quantities.

We remark that the previous analysis [CGH17] analyzed the first candidate *iO* [GGH+13b]. Whereas the work in [CGH17] heavily relies on the input partitionable property of the single input branching program, our algorithm do not need this property. Moreover, our algorithm can be applied to dual input branching program, so this attack can be applied to wider range of branching programs.

GMMSSZ.

Most notable result for BP obfuscation, GMMSSZ, is suggested by Garg *et al.* in TCC 2016 [GMM+16]. The authors claim the security of their construction against all known attack. Nevertheless, the matrix zeroizing attack can be applied to their obfuscation.

GMMSSZ obfuscates low-rank matrix branching program, which is evaluated by checking whether the product $\mathbf{M}_0 \cdot \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{M}_{\ell+1}$ is zero or not. There are two distinctive property of the obfuscation; the uniform random higher dimension embedding and given bookend vectors as inputs. Let $\mathbf{M}_0 = (\beta_1, \cdots, \beta_{d_1}), \mathbf{M}_{\ell+1} = (\gamma_1, \cdots, \gamma_{d_{\ell+1}})^T$ are the given bookend vectors. The bookend vectors are also extended as $\mathbf{H}_0 = (\mathbf{M}_0||\mathbf{0}), \mathbf{H}_{\ell+1} = (\mathbf{M}_{\ell+1}||\mathbf{U}_{\ell+1})^T$ for randomly chosen $\mathbf{U}_{\ell+1}$ in the higher dimension embedding step to remove the higher dimension embedding matrices. Note that the branching programs of this obfuscation are square, we do not restrict the shape of matrices in this section.

For the evaluation, one compute $\widetilde{\mathbf{M}}_0 \cdot \prod_{i\in[\ell]} \widetilde{\mathbf{M}}_{i,\mathbf{b}_i} \cdot \widetilde{\mathbf{M}}_{\ell+1}$, which is corresponding to

$$\mathbf{D_S} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T} = \rho_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i} \cdot \rho_{\mathbf{T}} \cdot \left(\mathbf{M}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{M}_{\ell+!} \right) \pmod{\langle \mathbf{g} \rangle}$$

in $\mathcal{R}$ program by Proposition 3.4.2. Since we know all $\mathbf{M}$'s, we can compute the ratios of scalar bundlings by

$$\rho_{j,\mathbf{b}_j}/\rho_{j,\mathbf{b}'_j} = \frac{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T}/\mathbf{M}_0 \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{M}_{\ell+1}}{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}'_i} \cdot \mathbf{D_T}/\mathbf{M}_0 \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}'_i} \cdot \mathbf{M}_{\ell+1}}$$

for $\mathbf{b}, \mathbf{b}'$ which are same at all but j-th bit. Therefore, the matrix zeroizing attack well works for the construction of [GMM+16]. We remark that this method works for *unknown* bookend matrices with more complicated technique, see Section 3.5.2.

3.5.2 Attackable BP Obfuscation, General Case

Now we consider the attackable BP obfuscations in general. We note that an attackable obfuscation without bookends can be considered as the obfuscation with bookends by re-naming the matrices. For example, if we name $\mathbf{D_S} := \mathbf{D}_{1,\mathbf{0}} = \rho_{1,\mathbf{0}} \cdot \mathbf{D}_1$, then we can regard that $\mathbf{D_S}$ is a left bookend matrix and $\rho_{1,\mathbf{0}}$ the corresponding scalar bundling.

The case of obfuscation with bookend matrices is most complex, and requires complicated technique. We will recover the bookend matrices up to constant multiplication, and proceed the algorithm similar to the case of [GMM+16].

Recovering the Bookends

For the sake of simplicity, we only consider the case of *bookend vectors*. To tackle constructions using bookend matrices, it is suffice to consider a fixed (u, v)-entry of output matrix given in Proposition 3.3.1.

If the obfuscation has bookend vectors, then the evaluation of $\mathcal{R}$ pro-

gram is computed by

$$\mathbf{D_S} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T} = \rho_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i} \cdot \rho_{\mathbf{T}} \cdot \left(\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0 \right) \pmod{\langle \mathbf{g} \rangle}$$

for some vectors $\mathbf{S}_0 \in (\mathcal{R}/\langle \mathbf{g} \rangle)^{1 \times d_1}$ and $\mathbf{T}_0 \in (\mathcal{R}/\langle \mathbf{g} \rangle)^{d_{\ell+1} \times 1}$. Let $\mathbf{S}_0 = (\beta_1, \cdots, \beta_{d_1})$, $\mathbf{T}_0 = (\gamma_1, \cdots, \gamma_{d_{\ell+1}})$ and the evaluation $\mathbf{D_S} \cdot \prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T}$ is denoted by $Eval_{\mathbf{D}}(\mathbf{b}_1, \cdots, \mathbf{b}_\ell)$.

Our idea is removing ρ's to make equations over $\mathbf{S}_0, \mathbf{T}_0$. Let $\mathbf{b}_{i,t} \in \{0,1\}^w$ for $1 \leqslant i \leqslant \ell$ and $t \in \{0,1\}$ and $\mathbf{t} = (t_1, \cdots, t_\ell) \in \{0,1\}^w$. Then the following two values share the same ρ's, precisely $(\rho_{\mathbf{S}}\rho_{\mathbf{T}})^2 \cdot \prod_{i \in [\ell]} \rho_{i,\mathbf{b}_{i,0}} \rho_{i,\mathbf{b}_{i,1}}$:

$$Eval_{\mathbf{D}}(\mathbf{b}_{1,0}, \cdots, \mathbf{b}_{\ell,0}) \cdot Eval_{\mathbf{D}}(\mathbf{b}_{1,1}, \cdots, \mathbf{b}_{\ell,1}),$$
$$Eval_{\mathbf{D}}(\mathbf{b}_{1,t_1}, \cdots, \mathbf{b}_{\ell,t_\ell}) \cdot Eval_{\mathbf{D}}(\mathbf{b}_{1,1-t_1}, \cdots, \mathbf{b}_{\ell,1-t_\ell}).$$

We denote $\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0$ by $Eqn_{\mathbf{M}}(\mathbf{b}_1, \cdots, \mathbf{b}_\ell)$. Then, by the above relations, we get a equation for $\beta_1, \cdots, \beta_{d_1}, \gamma_1, \cdots, \gamma_{d_{\ell+1}}$:

$$\frac{Eqn_{\mathbf{M}}(\mathbf{b}_{1,0}, \cdots, \mathbf{b}_{\ell,0}) \cdot Eqn_{\mathbf{M}}(\mathbf{b}_{1,1}, \cdots, \mathbf{b}_{\ell,1})}{Eval_{\mathbf{D}}(\mathbf{b}_{1,0}, \cdots, \mathbf{b}_{\ell,0}) \cdot Eval_{\mathbf{D}}(\mathbf{b}_{1,1}, \cdots, \mathbf{b}_{\ell,1})}$$
$$= \frac{Eqn_{\mathbf{M}}(\mathbf{b}_{1,t_1}, \cdots, \mathbf{b}_{\ell,t_\ell}) \cdot Eqn_{\mathbf{M}}(\mathbf{b}_{1,1-t_1}, \cdots, \mathbf{b}_{\ell,1-t_\ell})}{Eval_{\mathbf{D}}(\mathbf{b}_{1,t_1}, \cdots, \mathbf{b}_{\ell,t_\ell}) \cdot Eval_{\mathbf{D}}(\mathbf{b}_{1,1-t_1}, \cdots, \mathbf{b}_{\ell,1-t_\ell})}.$$

Both side of the equation is homogeneous polynomial of degree 4. If we substitute each degree 4 monomials by another variables, this equation become a homogeneous linear equation of new variables. The number of new variable is $O(d_1^2 d_{\ell+1}^2)$.

Now we assume that we can obtain sufficient number of linearly independent equations generated by the explained way. Then, since the system of linear equations can be solved in $O(M^3)$ time by Gaussian elimination

for the number of variable M, we can find all ratios of degree 4 monomials. [§] In other words, we can compute $\delta\beta_1, \cdots, \delta\beta_{d_1}, \delta\gamma_1, \cdots, \delta\gamma_{d_{\ell+1}}$ for some constant δ.

Matrix Zeroizing Attack

The remaining part of the attack is exactly same with the attack on GMMSSZ. Precisely, we can recover the ratios of scalar bundlings by computing

$$\rho_{j,\mathbf{b}_j}/\rho_{j,\mathbf{b}'_j} = \frac{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T}/\mathbf{S}_0 \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0}{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}'_i} \cdot \mathbf{D_T}/\mathbf{S}_0 \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}'_i} \cdot \mathbf{T}_0}$$

for $\mathbf{b}, \mathbf{b}'$ which are same at all but j-th bits. We note that we do not know exact values of $\mathbf{S}_0, \mathbf{T}_0$, but we recovered $\delta\mathbf{S}_0, \delta\mathbf{T}_0$ in the above step. Thus we can compute $\rho_{j,\mathbf{b}_j}/\rho_{j,\mathbf{b}'_j}$ by

$$\frac{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D_T}/(\delta\mathbf{S}_0) \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}_i} \cdot (\delta\mathbf{T}_0)}{\mathbf{D_S} \cdot \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}'_i} \cdot \mathbf{D_T}/(\delta\mathbf{S}_0) \prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}'_i} \cdot (\delta\mathbf{T}_0)}.$$

Therefore the matrix zeroizing attack can be applied to the attackable BP obfuscations, which include all existing BP obfuscations over GGH13.

[§]Here we assume that $\mathbf{g}$ is hard to factorize. If $\mathbf{g}$ is factorized in the Gaussian elimination procedure, we can proceed the algorithm for a factor of $\mathbf{g}$.

Chapter 4

Mathematical Analysis of Indistinguishability Obfuscation based on the GGH15 Multilinear Map

In this chapter, we present a new cryptanalytic algorithm on obfuscations based on GGH15 multilinear map. Our algorithm, *statistical zeroizing attack*, directly distinguishes two distributions from obfuscation while it follows the zeroizing attack paradigm, that is, it uses evaluations of zeros of obfuscated programs.

Our attack breaks the recent indistinguishability obfuscation candidate suggested by Chen *et al.* (CRYPTO'18) for the optimal parameter settings. More precisely, we show that there are two functionally equivalent branching programs whose CVW obfuscations can be efficiently distinguished by computing the sample variance of evaluations.

This statistical attack gives a new perspective on the security of the indistinguishability obfuscations: we should consider the shape of the distributions of evaluation of obfuscation to ensure security.

In other words, while most of the previous (weak) security proofs have been studied with respect to algebraic attack model or ideal model, our attack shows that this algebraic security is not enough to achieve indistinguishability obfuscation. In particular, we show that the obfuscation scheme suggested by Bartusek *et al.* (TCC'18) does not achieve the desired security in a certain parameter regime, in which their algebraic security proof still holds.

The correctness of statistical zeroizing attacks holds under a mild assumption on the preimage sampling algorithm with a lattice trapdoor. We experimentally verify this assumption for implemented obfuscation by Halevi *et al.* (ACM CCS'17).

4.1 Preliminaries

4.1.1 Notations

Throughout this chapter, lower bold letters means row vectors and capital bold letters denote matrices. In addition, capital italic letters denote random matrices or random variables. For a random variable X, we let $E(X)$ be the expected value of X, $Var(X)$ the variance of X.

The n-dimensional identity matrix is denoted by $\mathbf{I}^{n\times n}$. For a row vector $\mathbf{v}$, a i-th component of $\mathbf{v}$ is denoted by v_i, and for a matrix $\mathbf{A}$, a (i,j)-th entry of a matrix $\mathbf{A}$ is denoted by $a_{i,j}$, respectively. A notation $\mathbf{1}^{a\times b}$ means a $a \times b$ matrix such that all entries are 1. The ℓ_p norm of a vector $\mathbf{v} = (v_i)$ is denoted by $\|\mathbf{v}\|_p = (\sum_i |v_i|^p)^{1/p}$. We denote $\|\mathbf{A}\|_\infty$ by the infinity norm of

a matrix $\mathbf{A}$, $\|\mathbf{A}\|_\infty = \max_{i,j} a_{i,j}$ with $\mathbf{A} = (a_{i,j})$.

4.2 Statistical Zeroizing Attack

In this section, we introduce a new cryptanalysis, *statistical zeroizing attack*. We give an abstract model for branching program obfuscation and the attack description in this model. In this attack, we are given two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$, which will be specified later, and an obfuscated program $\mathcal{O}(\mathbf{P})$ for $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. Our purpose is to distinguish whether $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. The targeted branching programs of the obfuscation output 0 when the product corresponding to input is zero. The obfuscated program $\mathcal{O}(\mathbf{P})$ consists of

$$\left\{\mathbf{S}, \{\mathbf{D}_{i,\mathbf{b}}\}_{1\leqslant i\leqslant h, \mathbf{b}\in\{0,1\}^s}, \mathbf{T}, \mathsf{inp} = (\mathsf{inp}_1, \cdots, \mathsf{inp}_s) : [h] \to [\ell]^s, B\right\}$$

where every element is a matrix over $\mathbb{Z}_q$ (possibly identity) except the input function inp. The output of the obfuscated program at $\mathbf{x} = (x_1, \cdots, x_\ell) \in \{0,1\}^\ell$ is computed by considering the value

$$\mathcal{O}(\mathbf{P})(\mathbf{x}) = \mathbf{S} \cdot \prod_{i=1}^{h} \mathbf{D}_{i,\mathbf{x}_{\mathsf{inp}}(i)} \cdot \mathbf{T}$$

where $\mathbf{x}_{\mathsf{inp}(i)} = (x_{\mathsf{inp}_1(i)}, \cdots, x_{\mathsf{inp}_s(i)})$. Note that $\mathcal{O}(\mathbf{P})(\mathbf{x})$ can be a matrix, vector or an element (over $\mathbb{Z}_q$). Regard it as matrix/vector/integer over $\mathbb{Z}$ and check the value: if $\|\mathcal{O}(\mathbf{P})(\mathbf{x})\|_\infty < B < q$ then it outputs 0, otherwise outputs 1. We call $\mathcal{O}(\mathbf{P})(\mathbf{x})$ the *evaluation* of the obfuscated program (at $\mathbf{x}$). We also call $\mathcal{O}(\mathbf{P})(\mathbf{x})$ evaluation of zero if $\mathbf{P}(\mathbf{x}) = 0$ in the plain program. We stress that the *output* and *evaluation* of the obfuscated program

is different; the output of the obfuscated program is the same to output of original program, and the evaluation is the value $\mathcal{O}(\mathbf{P})(\mathbf{x})$, which is computed right before determining the output.

To distinguish two different obfuscated programs, we see the distribution of valid evaluations of zero of $\mathcal{O}(\mathbf{M})$ and $\mathcal{O}(\mathbf{N})$. For the evaluation of zero, the size of these products is far smaller than q (or B), thus we can obtain the integer value rather than the element in $\mathbb{Z}_q$. Now, if the evaluation is of the matrix or vector form, we consider only the first entry, namely $(1, 1)$ entry of the matrix or the first entry of the vector, in the whole procedure of the attack. We call all of these entries by *the first entry* of the evaluation, including the case of the evaluation is just a real value.

Our strategy is to compute the sample variance of the first entries of many independent evaluations which follow the same distribution. The key of the attack is that this variance heavily depends on the plain program of the obfuscated program and the variance is sufficiently different to distinguish for two certain programs. Therefore, from the variance of the several evaluations, we can decide that the obfuscated program is from which program.

Note that one can sample an element following the distribution of obfuscation or its evaluation at fixed point $\mathbf{x} = \mathbf{x}_0$ in polynomial time when the corresponding program is given, since there is no private key in the obfuscation procedure. In this regard, we consider a more general problem which is easier to analyze: Given two polynomial-time constructible distribution $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ and x sampled from one of them, determine that the sample is from which distribution. In our scenario, $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ are the distribution of $\mathcal{O}(\mathbf{M})(\mathbf{x})$ and $\mathcal{O}(\mathbf{N})(\mathbf{x})$, respectively where the distribution is over all randomness to construct obfuscations.

Since the adversary has one sample in our setting, the actual algorithm proceeds by sampling multiple evaluations itself as follows.

Data: $\mathcal{D}_{\mathbf{M}}, \mathcal{D}_{\mathbf{N}}, x, \kappa$

1. set $B = (\sigma_{\mathbf{M}}^2 + \sigma_{\mathbf{N}}^2)/2$ for $\sigma_{\mathbf{M}}^2 = Var(\mathcal{D}_{\mathbf{M}})$ and $\sigma_{\mathbf{N}}^2 = Var(\mathcal{D}_{\mathbf{N}})$
2. $i \leftarrow [\kappa]$ and let $s_i = x$
3. sample $\{s_j\}_{j\in[i-1]}$ from $\mathcal{D}_{\mathbf{M}}$ and $\{s_j\}_{i+1\leqslant j\leqslant\kappa}$ from $\mathcal{D}_{\mathbf{N}}$
4. compute the sample variance S^2 of $\{s_j\}_{j\in[\kappa]}$
5. if $S^2 < B$, decides $\mathcal{D}_{\mathbf{M}}$, otherwise $\mathcal{D}_{\mathbf{N}}$.

The choice of κ is specified later in Proposition 4.2.1. We also remark that the overall time complexity of algorithm is $O(\kappa \cdot T_{\mathsf{sample}})$ plus small computation for sample variance, where T_{sample} is the time complexity for sampling algorithms. The advantage of this algorithm is, by the standard hybrid arguemnt, $\mathsf{adv}_{\mathsf{mult}}/\kappa$ where $\mathsf{adv}_{\mathsf{mult}} = 0.98$ is the advantage of distinguishing algorithm by sample variance when κ samples are given as inputs instead of one sample as in Proposition 4.2.1.

In the next subsection, we analyze the distinguishing algorithm using sample variance for general distributions instead of iO when the multiple samples are given. Then we go back to the actual attack for iO for the concrete obfuscations in Section 4.3 and 4.4 by showing the attack conditions hold well.

4.2.1 Distinguishing Distributions using Sample Variance

Now we give the detailed analysis of distinguishing by sample variance. In this algorithm, we compute the variance of the samples, and check whether the distance between the sample variance and the expected variance of $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$. If the distance from the sample variance to the variance of $\mathcal{D}_{\mathbf{M}}$ is less than the distance to the variance of $\mathcal{D}_{\mathbf{N}}$, we decide the given samples are from $\mathcal{D}_{\mathbf{M}}$. Otherwise we decide the samples are from $\mathcal{D}_{\mathbf{N}}$. The result of this method is stated in the following proposition.

Proposition 4.2.1. *Suppose that two random variables $X_{\mathbf{M}}$ and $X_{\mathbf{N}}$ that follow polynomial time constructible distributions $\mathcal{D}_{\mathbf{N}}$ and $\mathcal{D}_{\mathbf{M}}$ and have the means $\mu_{\mathbf{M}}$ and $\mu_{\mathbf{N}}$ and the variances $\sigma^2_{\mathbf{N}}$ and $\sigma^2_{\mathbf{M}}$, respectively. For the security parameter λ and polynomials $p, q, r = \mathsf{poly}(\lambda)$, there is a polynomial time algorithm that distinguishes $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ with non-negligible advantage when $O(p \cdot (\sqrt{q} + \sqrt{r})) = poly(\lambda)$ independent samples from $\mathcal{D}_{\mathbf{P}}$ are given and the following conditions hold:*

$$\left|\frac{\max(\sigma^2_{\mathbf{N}}, \sigma^2_{\mathbf{M}})}{\sigma^2_{\mathbf{N}} - \sigma^2_{\mathbf{M}}}\right| \leqslant p \ \left|\frac{E[(X_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma^4_{\mathbf{N}}}\right| \leqslant q, \text{ and } \left|\frac{E[(X_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma^4_{\mathbf{M}}}\right| \leqslant r.$$

In other words, if two known distributions satisfy the conditions, we can solve the distinguishing problem of two distribution with multiple samples. Thus to cryptanalyze the concrete obfuscation schemes, it suffice to show the conditions in Proposition 4.2.1. We conclude this section by giving the proof of this proposition.

Proposition 4.2.1. We call a definition and useful lemmas first.

Lemma 4.2.1 (Chebyshev's inequality)**.** *Let X be a random variable with a finite expected value μ and a finite variance $\sigma^2 > 0$. Then, it holds that*

$$\Pr[|X - \mu| \geqslant k\sigma] \leqslant 1/k^2$$

for any real number $k > 0$.

Definition 4.2.1 (Sample variance)**.** *Given random n samples $x_1, x_2, \cdots, x_n$ of $\mathcal{D}$, the sample variance of $\mathcal{D}$ is defined by*

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean.

Definition 4.2.2 (Kurtosis)**.** *Let X be a random variable with a finite expected value μ and a finite variance $\sigma^2 > 0$. The kurtosis of X is defined by*

$$Kurt[X] = \frac{E[(X-\mu)^4]}{E[(X-\mu)^2]^2} = \frac{E[(X-\mu)^4]}{\sigma^4}.$$

Lemma 4.2.2. *Let S^2 be the sample variance of size κ samples of a distribution $\mathcal{D}$. Let X be a random variable following $\mathcal{D}$ and $\mu_n = E[(X - E[X])^n]$ be the n-th central moment. Then the variance of S^2 satisfies*

$$Var(S^2) = \frac{1}{\kappa}\left(\mu_4 - \frac{\kappa - 3}{\kappa - 1}\mu_2^2\right).$$

Now we return to the proof. Suppose that all of the conditions hold for polynomials $p, q, r \in \mathsf{poly}(\lambda)$ and $\sigma_{\mathbf{M}}^2 < \sigma_{\mathbf{N}}^2$. By Lemma 4.2.1 and 4.2.2, we

compute the 99% confidence interval of variance of S^2 as follows

$$\Pr\left[|S^2 - \sigma_{\mathbf{P}}^2| \geqslant 10 \cdot \sqrt{\frac{1}{\kappa} \cdot \left(E[(X_{\mathbf{P}} - \mu_{\mathbf{P}})^4] - \frac{\kappa - 1}{\kappa - 3} \cdot \sigma_{\mathbf{P}}^4\right)}\right] \leqslant \frac{1}{100}$$

with κ number of samples. If κ is sufficiently large, the two intervals of sample variance for $\mathbf{M}$ and $\mathbf{N}$ are disjoint. So we can distinguish two distributions by checking the size of sample variance.

More precisely, if $\kappa \geqslant 100 \cdot (p \cdot \sqrt{q} + p \cdot \sqrt{r})^2$ that is $\mathsf{poly}(\lambda)$, we have

$$\begin{aligned}\sigma_{\mathbf{M}}^2 + 10 \cdot \sigma_{\mathbf{M}}^2 \cdot \sqrt{\frac{1}{\kappa} \cdot \left(\frac{E[(X_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma_{\mathbf{M}}^4} - \frac{\kappa - 1}{\kappa - 3}\right)} \\ < \sigma_{\mathbf{N}}^2 - 10 \cdot \sigma_{\mathbf{N}}^2 \cdot \sqrt{\frac{1}{\kappa} \cdot \left(\frac{E[(X_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma_{\mathbf{N}}^4} - \frac{\kappa - 1}{\kappa - 3}\right)}\end{aligned}$$

Thus the algorithm decides the answer by checking if the sample variance is included in which interval; we do not care the case that it is not included both. This algorithm succeeds with probability at least 0.99 for each input, i.e. the advantage of algorithm is at least 0.98. Note that this algorithm only does the polynomial number of sampling and computing the variance, thus the running time is polynomial. □

4.3 Cryptanalysis of CVW Obfuscation

In this section, we briefly describe the construction of CVW obfuscation scheme and show that the statistical zeroizing attack works well for CVW obfuscation.

4.3.1 Construction of CVW Obfuscation

Chen, Vaikuntanathan and Wee proposed a new candidate of iO which is robust against all existing attacks. We here give a brief description of the candidate scheme. For more details, we refer to original paper [CVW18].

First, we start with the description of BPs they used. The authors use single-input binary BPs, *i.e.*, $\mathsf{inp} = \mathsf{inp}_1$. They employ a new function, called an input-to-index map $\bar{\omega}$: $\{0,1\}^\ell \to \{0,1\}^h$ such that $\bar{\omega}(\mathbf{x})_i = \mathbf{x}_{\mathsf{inp}(i)}$ for all $i \in [h]$, $\mathbf{x} \in \{0,1\}^\ell$. As used in the paper [CVW18], we denote the $\prod_{i=1}^{h} \mathbf{M}_{i,\bar{\omega}(\mathbf{x})_i}$ by $\mathbf{M}_{\bar{\omega}(\mathbf{x})}$ or simply $\mathbf{M}_{\mathbf{x}}$. We sometimes abuse the notion $\mathbf{M}_{i,x_i}$ to denote $\mathbf{M}_{i,\bar{\omega}(\mathbf{x})_i}$.

A target BP $\mathbf{P} = \{\mathsf{inp}, \{\mathbf{P}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathcal{P}_0, \mathcal{P}_1\}$, which is called *Type I* BP in the original paper, satisfies the following conditions.

1. All the matrices $\mathbf{P}_{i,b}$ are $w \times w$ matrices.
2. For a vector $\mathbf{v} = \mathbf{1}^{1\times w}$, the target sets $\mathcal{P}_0, \mathcal{P}_1$ satisfies $\mathbf{v}\cdot\mathcal{P}_0 = \{\mathbf{0}^{1\times w}\}$, $\mathbf{v}\cdot\mathcal{P}_1 \neq \{\mathbf{0}^{1\times w}\}$.*
3. An index length h is set to $(\lambda+1)\cdot\ell$ with the security parameter λ.
4. An index-to-input function satisfies $\mathsf{inp}(i) = (i \mod \ell)$. Thus, index-to-input function iterates $\lambda+1$ times.

Construction. CVW obfuscation is a probabilistic polynomial time algorithm which takes as input a BP $\mathbf{P}$ with an input length ℓ, and outputs an obfuscated program preserving the functionality. The algorithm process consists of the following steps. Here we use new parameters $n, m, q, t :=$

*As noted in the remark of introduction, it is assumed implicitly that $\mathbf{v} = \mathbf{1}^{1\times w}$ for the targeted BP, while the definition of Type I BP uses $\mathbf{v} \in \{0,1\}^{1\times w}$.

$(w+2n\ell)\cdot n, \sigma$ for the construction. We will specify the parameter settings later.

- Sample bundling matrices $\{\mathbf{R}_{i,b} \in \mathbb{Z}^{2n\ell\times 2n\ell}\}_{i\in[h],b\in\{0,1\}}$ such that $(\mathbf{1}^{1\times 2\ell}\otimes \mathbf{I}^{n\times n})\cdot\mathbf{R}_{\mathbf{x}'}\cdot(\mathbf{1}^{2\ell\times 1}\otimes\mathbf{I}^{n\times n}) = \mathbf{0} \iff \mathbf{x}' \in \bar{\omega}(\{0,1\}^{\ell})$ for all $\mathbf{x}' \in \{0,1\}^h$. More precisely, $\mathbf{R}_{i,b}$ is a block diagonal matrix $\mathsf{diag}(\mathbf{R}_{i,b}^{(1)}, \mathbf{R}_{i,b}^{(2)}, \cdots, \mathbf{R}_{i,b}^{(\ell)})$. Each $\mathbf{R}_{i,b}^{(k)} \in \mathbb{Z}^{2n\times 2n}$ is one of the following three cases.

$$\mathbf{R}_{i,b}^{(k)} = \begin{cases} \mathbf{I}^{2n\times 2n} & \text{if } \mathsf{inp}(i) \neq k \\ \begin{pmatrix} \tilde{\mathbf{R}}_{i,b}^{(k)} & \\ & \mathbf{I}^{n\times n} \end{pmatrix}, \tilde{\mathbf{R}}_{i,b}^{(k)} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n} & \text{if } \mathsf{inp}(i) = k \text{ and } i \leqslant \lambda\ell \\ \begin{pmatrix} -\mathbf{I}^{n\times n} & \\ & \prod_{j=0}^{\lambda-1} \tilde{\mathbf{R}}_{k+j\ell,b}^{(k)} \end{pmatrix} & \text{if } \mathsf{inp}(i) = k \text{ and } i > \lambda\ell \end{cases}$$

- Sample matrices $\{\mathbf{S}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}\}_{i\in[h],b\in\{0,1\}}$ and compute

$$\mathbf{J} := (\mathbf{1}^{1\times(w+2n\ell)} \otimes \mathbf{I}^{n\times n}) \in \mathbb{Z}^{n\times t}$$
$$\hat{\mathbf{S}}_{i,b} := \begin{pmatrix} \mathbf{P}_{i,b}\otimes\mathbf{S}_{i,b} & \\ & \mathbf{R}_{i,b}\otimes\mathbf{S}_{i,b} \end{pmatrix} \in \mathbb{Z}^{t\times t}$$
$$\mathbf{L} := (\mathbf{1}^{(w+2n\ell)\times 1} \otimes \mathbf{I}^{n\times n}) \in \mathbb{Z}^{t\times n}$$

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leqslant i \leqslant h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n\times n})$, $\{\mathbf{E}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{t\times m}\}_{i\in[h-1],b\in\{0,1\}}$ and $\{\mathbf{E}_{h,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{t\times n}\}_{b\in\{0,1\}}$.

- Run Sample algorithms to obtain

$$\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ for } 1 \leqslant i \leqslant h-1,$$
$$\mathbf{D}_{h,b} \in \mathbb{Z}^{m\times n} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \hat{\mathbf{S}}_{h,b} \cdot \mathbf{L} \cdot \mathbf{A}_h + \mathbf{E}_{h,b}, \sigma).$$

- Define $\mathbf{A}_{\mathbf{J}}$ as a matrix $\mathbf{J} \cdot \mathbf{A}_0 \in \mathbb{Z}^{n\times m}$ and outputs matrices

$$\left\{\mathsf{inp}, \mathbf{A}_{\mathbf{J}}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}\right\}.$$

Evaluation. Evaluation process consists of two steps. The first step is to compute a matrix $\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})} \bmod q$. The last step is size comparison: If $\|\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})} \bmod q\|_\infty \leqslant B$, output 0 for some fixed B. Otherwise, output 1.

Parameters. Let λ and λ_{LWE} for the security parameters of obfuscation itself and underlying LWE problem satisfying $\lambda_{LWE} = \mathsf{poly}(\lambda)$ and the following constraints. Set $n = \Omega(\lambda_{LWE} \log q)$ and $\chi = D_{\mathbb{Z},2\sqrt{\lambda_{LWE}}}$. Moreover, for the trapdoor functionality, $m = \Omega(t \log q)$ and $\sigma = \Omega(\sqrt{t \log q})$ for $t = (w + 2n\ell) \cdot n$. $B \geqslant (w + 2n\ell) \cdot h \cdot (m \cdot \sigma^2\sqrt{n(w + 2n\ell)\sigma})^h$ and $q = B \cdot \omega(\mathsf{poly}(\lambda))$ for correctness, and $q \leqslant (\sigma/\lambda_{LWE}) \cdot 2^{\lambda_{LWE}^{1-\epsilon}}$ for a fixed $\epsilon \in (0, 1)$ for security. For more details, we refer readers to the original paper [CVW18].

Remark 4.3.1. *The original paper [CVW18] only uses one security parameter λ, but the correctness does not hold in that setting. Instead, the trick that uses two security parameters λ and λ_{LWE} resolves this problem as in [BGMZ18].*

Zerotest Functionality. From the construction of the obfuscation, the

following equality always holds, which is essentially what we need.

$$[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q = \left[\mathbf{J} \cdot \left(\prod_{i=1}^{h} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{A}_h + \mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right]_q$$

The honest evaluation with $\mathbf{P}_{\mathbf{x}} = \mathbf{0}^{w \times w}$ gives $\hat{\mathbf{S}}_{\mathbf{x}} = \mathbf{0}^{t \times t}$ due to the construction of $\mathbf{R}_{i,b}$ is zero for the valid evaluation. Then, the following inequality holds:

$$\begin{aligned}
\|[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q\|_\infty &= \left\| \left[\mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right]_q \right\|_\infty \\
&\leqslant \left\| \mathbf{J} \cdot \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right\|_\infty \\
&\leqslant h \cdot \left(\max_{i,b} \|\hat{\mathbf{S}}_{i,b}\| \cdot \sigma \cdot m \right)^h \leqslant B
\end{aligned}$$

for all but negligible probability due to the choice of B. If $\mathbf{P}_{\mathbf{x}}$ is not the zero matrix, then $\hat{\mathbf{S}}_{\mathbf{x}}$ is also not the zero matrix with overwhelming probability. It implies that $\|[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}_{\bar{\omega}(\mathbf{x})}]_q\|_\infty$ is larger than B with overwhelming probability because of $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n \times n})$.

4.3.2 Cryptanalysis of CVW Obfuscation

We apply the statistical zeroizing attack to the CVW obfuscation. As stated in Section 4.2, it is enough to show that the conditions of Proposition 4.2.1 hold. We only consider small variance σ^2 so that $\sigma = \mathsf{poly}(\lambda)$, and sufficiently large ℓ.[†] This includes the optimal parameter choice as

[†]Indeed, the attack requires the condition $\sigma^4 < m^\ell / n^{\ell+1}$.

well.

Our targeted two functionally equivalent BPs $\mathbf{M} = \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ and $\mathbf{N} = \{\mathbf{N}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ are of the form

$$\mathbf{M}_{i,b} = \mathbf{0}^{w\times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,b} = \begin{cases} \mathbf{1}^{w\times w} & \text{if } i = 1 \\ \mathbf{0}^{w\times w} & \text{otherwise} \end{cases}.$$

Suppose that we have an obfuscated program $\mathcal{O}(\mathbf{P})$ for $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. Our goal is to determine whether the program $\mathcal{O}(\mathbf{P})$ is an obfuscation of $\mathbf{M}$ or $\mathbf{N}$.

By the standard hybrid argument, it suffices to distinguish the distributions $\mathcal{D}_{\mathbf{M}}$ or $\mathcal{D}_{\mathbf{N}}$ where $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ is the distributions of the (1,1) entry of evaluation at a fixed vector $\mathbf{x}$ of the obfuscated program of $\mathbf{M}$ or $\mathbf{N}$, respectively. To exploit Proposition 4.2.1, we transform the CVW construction into the language of random variables. We denote the random matrix by the capital italic words whose entry follows a distribution that corresponds to the distribution of entry of the bold matrix. For example, the entry of random matrix $E_{i,b}$ follows the distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ since the matrix $\mathbf{E}_{i,b}$ is chosen from $\mathcal{D}_{\mathbb{Z},\sigma}^{t\times m}$ in the CVW construction. More precisely, we define random matrices $\tilde{R}_{i,b}^{(k)}$ following $\mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}$, $S_{i,b}$ following $\mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}$ and A_i as in the trapdoor sampling algorithm. Then we obtain random matrices $\hat{S}_{i,b}^{(\mathbf{P})}$, $R_{i,b}^{(\mathbf{P})}$, $E_{i,b}^{(\mathbf{P})}$ and $D_{i,b}^{(\mathbf{P})}$ as in the construction of CVW obfuscation for the branching programs $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. We note that only $\hat{S}_{i,b}^{(\mathbf{P})}$ and $D_{i,b}^{(\mathbf{P})}$ depend on the choice of branching program, but we put $\mathbf{P}$ in some other random variables for convenience of distinction.

Under this setting, it suffices to show the following proposition.

Proposition 4.3.1. *For a security parameter λ, fix the Gaussian variance*

parameter $\sigma = \mathsf{poly}(\lambda)$*. Then, there are two functionally equivalent branching programs* $\mathbf{M}$ *and* $\mathbf{N}$ *with sufficiently large input length* ℓ *satisfying the following statement: let* $Z_{\mathbf{M}}$ *and* $Z_{\mathbf{N}}$ *be random variables satisfying*

$$Z_{\mathbf{M}} = \left[\left(\mathbf{J} \cdot A_0 \cdot D^{(\mathbf{M})}_{\bar{\omega}(\mathbf{x})} \right)_{(1,1)} \right]_q, \ Z_{\mathbf{N}} = \left[\left(\mathbf{J} \cdot A_0 \cdot D^{(\mathbf{N})}_{\bar{\omega}(\mathbf{x})} \right)_{(1,1)} \right]_q$$

where every random matrix is defined as the above. Let $\mu_{\mathbf{M}}$ *and* $\mu_{\mathbf{N}}$*,* $\sigma^2_{\mathbf{M}}$ *and* $\sigma^2_{\mathbf{N}}$ *be mean and variance of the random variables of* $Z_{\mathbf{M}}$ *and* $Z_{\mathbf{N}}$*, respectively. Then, it holds that*

$$\left| \frac{\max(\sigma^2_{\mathbf{N}}, \sigma^2_{\mathbf{M}})}{\sigma^2_{\mathbf{N}} - \sigma^2_{\mathbf{M}}} \right| \leqslant p, \ \left| \frac{E[(Z_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma^4_{\mathbf{N}}} \right| \leqslant q, \text{ and } \left| \frac{E[(Z_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma^4_{\mathbf{M}}} \right| \leqslant q.$$

for some $p, q = \mathsf{poly}(\lambda)$ *under Assumption 1.*

We remark that since the random matrices D's are dependent each other, we need to assume the statistical property for verifying conditions of Proposition 4.3.1 as follows.

Assumption 1. For an integer $0 \leqslant k \leqslant h-2$ and $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$, let $\hat{D}^{(\mathbf{P})}_k$ be a random matrix such that $\hat{D}^{(\mathbf{P})}_k = \prod_{i=k+2}^{h} D^{(\mathbf{P})}_i$, where $D^{(\mathbf{P})}_i$ is the random matrix which follows a distribution corresponding preimage-sampled matrix $\mathbf{D}^{(\mathbf{P})}_i$. Then, the following equations hold

1. the variance is approximated by the same one assumed that D's are independent Gaussian, that is, it holds that

$$Var[\hat{D}^{(\mathbf{P})}_k] = \Theta\left(m^{h-k-2}(\sigma^2)^{h-k-1}\right).$$

2. the kurtosis is bounded by constant, that is, it holds that

$$\frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} = O(\mathsf{poly}(\lambda)).$$

We experimentally verify this assumption using the implementation of GGH15 BP obfuscation by Halevi *et al.* [HHSSD17a]. More detailed experimental results are presented in Appendix 6.2.5. We remark that if we assume that D's are independent matrices that have discrete Gaussian entry with the variance σ^2, the following computations hold:

- the variance of $\hat{D}_k^{(\mathbf{P})}$ is exactly $m^{h-k-2} \cdot (\sigma^2)^{h-k-1}$, and
- the kurtosis of $\hat{D}_k^{(\mathbf{P})}$ is $3 \cdot (1 + 2/m)^{h-k} = \Theta(1)$.

The honest evaluation of the CVW obfuscation $[\mathbf{A}_{\mathbf{J}} \cdot \mathbf{D}^{(\mathbf{P})}_{\bar{\omega}(\mathbf{x})}]_q$ is the matrix of the form

$$\mathbf{J} \cdot \sum_{j=0}^{h-1} \left(\left(\prod_{i=1}^{j} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j+1,x_{j+1}} \cdot \prod_{k=j+2}^{h} \mathbf{D}^{(\mathbf{P})}_{k,x_k} \right),$$

which does not contain the term including the trapdoor matrices $\mathbf{A}_i$ for $i = 0, \cdots, h-1$. Thus, to establish the statistical properties including variance in Proposition 4.3.1, it suffices to analyze the statistical properties of the random matrices $\hat{S}^{(\mathbf{P})}_{i,b}$, $E^{(\mathbf{P})}_{i,b}$, $D^{(\mathbf{P})}_{i,b}$ and their products.

By the definition of $Z_{\mathbf{P}}$ with $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$, it is rewritten as

$$Z_{\mathbf{P}} = \mathbf{J} \cdot \sum_{j=0}^{h-1} \left(\left(\prod_{i=1}^{j} \hat{S}_{i,x_i} \right) \cdot E_{j+1,x_{j+1}} \cdot \prod_{k=j+2}^{h} D^{(\mathbf{P})}_{k,x_k} \right).$$

Now we give the lemmas to prove Proposition 4.3.1. The proofs of lemmas are placed in Appendix 6.2.7 and sub-lemmas in Appendix 6.2.6.

The proof of Proposition 4.3.1 using the lemmas is placed in the concluding part of this section.

For the convenience of the statement, let $(Z_{1,1}^{(\mathbf{M})})_j$ be random variables of $(1,1)$-th entry of the random matrices

$$\mathbf{J} \cdot \prod_{i=1}^{j} \hat{S}_i^{(\mathbf{M})} \cdot E_{j+1}^{(\mathbf{M})} \cdot \prod_{k=j+2}^{h} D_k^{(\mathbf{M})}$$

for $j = 0, 1, \cdots, h-1$. In this notation, $Z_{\mathbf{M}}$ is the summation of $(Z_{1,1}^{(\mathbf{M})})_j$ for $j \in \{0, 1, \cdots, h-1\}$. Similarly, we define $(Z_{1,1}^{(\mathbf{N})})_j$ for all $j = 0, \cdots, h-1$. We employ additional notations constants c, d and (possibly polynomial) c_0 such that for all $0 \leqslant k \leqslant h-2$,

$$c \leqslant \frac{Var[\hat{D}_k^{(\mathbf{P})}]}{m^{h-k-2}(\sigma^2)^{h-k-1}} \leqslant d \quad \text{and} \quad \frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} \leqslant c_0.$$

We remark that variances of many terms for $\mathbf{M}$ and $\mathbf{N}$ are *exactly the same* since the only D_1, $\hat{S}_1$ are different and the different terms in products of $\hat{S}$ are canceled for $j \geqslant 2$. Note that most of lemmas hold under Assumption 1, but we omit this repeated statement *under Assumption 1* for brevity.

Lemma 4.3.1. *$E[(Z_{1,1}^{(\mathbf{M})})_j] = E[(Z_{1,1}^{(\mathbf{N})})_j] = 0$ for all $j = 0, \cdots, h-1$.*

Lemma 4.3.2. *$E[(Z_{1,1}^{(\mathbf{M})})_{\mu_1} \cdot (Z_{1,1}^{(\mathbf{M})})_{\mu_2}] = E[(Z_{1,1}^{(\mathbf{N})})_{\mu_1} \cdot (Z_{1,1}^{(\mathbf{N})})_{\mu_2}] = 0$ for $\mu_1 \neq \mu_2$.*

Lemma 4.3.3 ($j = 0$)**.** *It holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_0] = Var[(Z_{1,1}^{(\mathbf{N})})_0] = \Theta\left((w + 2n\ell) \cdot m^{h-1} \cdot \sigma^{2h}\right) \text{ and}$$
$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_0^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_0]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_0^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_0]^2}\right| \leqslant 3c_0 \cdot (w + 2n\ell)^2 \cdot m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.3.4 ($j = 1$)**.** *It holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_1] = \Theta\left(\left(n^3\sigma^2 + (2\ell - 1) \cdot n^2\right) \cdot m^{h-2}(\sigma^2)^h\right),$$
$$Var[(Z_{1,1}^{(\mathbf{N})})_1] = \Theta\left(w^3 \cdot n \cdot m^{h-2}(\sigma^2)^h\right) + Var[(Z_{1,1}^{(\mathbf{M})})_1]$$
$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_1]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_1]^2}\right| \leqslant 27c_0 \cdot (w + 2n\ell)^4 n^2 m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.3.5 ($1 < j \leqslant \lambda \cdot \ell$)**.** *Let j be a fixed integer with $j = \ell \cdot j_1 + j_2 > 1$ for $0 \leqslant j_2 < \ell$ and $2 \leqslant j \leqslant \lambda \cdot \ell$. Then, it holds that*

$$Var[(Z_{1,1}^{(\mathbf{M})})_j] = Var[(Z_{1,1}^{(\mathbf{N})})_j]$$
$$= \Theta\left(\left(j_2 n^{j+j_1+2}(\sigma^2)^{j_1+1} + (\ell - j_2)n^{j+j_1+1}(\sigma^2)^{j_1} + \ell n^{j+1}\right)m^{h-j-1}(\sigma^2)^h\right).$$

Moreover, it holds that

$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_j]^2}\right| \leqslant 27c_0(w + 2n\ell)^4 n^2 m^2 \left(1 + \frac{2}{n}\right)^{j_1+j-1} \left(\frac{d}{c}\right)^2$$
$$= \mathsf{poly}(\lambda).$$

Lemma 4.3.6 ($j > \lambda \cdot \ell$))**.** *Let j be a fixed integer with $j = \ell \cdot j_1 + j_2 > 1$*

for $0 \leqslant j_2 < \ell$ *and* $j > \lambda \cdot \ell$. *Then, it holds that*

$$\begin{aligned} &Var[(Z_{1,1}^{(\mathbf{M})})_j] = Var[(Z_{1,1}^{(\mathbf{N})})_j] \\ &= \Theta\left(\left((\ell + j_2) \cdot n^{\lambda+j+1} \cdot (\sigma^2)^\lambda + (\ell - j_2) \cdot n^{j+1}\right) \cdot m^{h-j-1} \cdot (\sigma^2)^h\right). \end{aligned}$$

In addition, it holds that

$$\begin{aligned} \left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_j]^2}\right| &\leqslant 27c_0(w + 2n\ell)^4 n^2 m^2 \left(1 + \frac{2}{n}\right)^{\lambda+j-2} \left(\frac{d}{c}\right)^2 \\ &= \mathsf{poly}(\lambda). \end{aligned}$$

Now we give a proof of the proposition 4.3.1 using above lemmas.

of Proposition 4.3.1. Fix ℓ be a sufficiently large so that $\sigma^4 < m^\ell / n^{\ell+1}$ and choose BP $\mathbf{M}$ and $\mathbf{N}$ as the given in the first page of this section. These two branching programs have the same functionality and length.

Using the results of lemmas, we can prove the proposition by analyzing the summation of random matrices. We first verify the results for $Z_{\mathbf{M}}$. The similar result holds for $Z_{\mathbf{N}}$ since the bounds of lemmas are almost same.

From Lemma 4.3.1, 4.3.2 and the definition of $Z_{\mathbf{M}}$, we have

$$Var[Z_{\mathbf{M}}] = E\left[(\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j)^2\right] = E\left[\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^2\right] = \sum_{j=0}^{h-1} Var[(Z_{1,1}^{(\mathbf{M})})_j].$$

On the other hands, applying to the Cauchy-Schwarz inequality, it also holds

$$E[Z_{\mathbf{M}}^4] = E\left[(\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j)^4\right] \leqslant E\left[h^3 \cdot (\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4)\right].$$

When dividing both sides by $Var[Z_{\mathbf{M}}]^2$, we obtain the inequality

$$\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| \leqslant \left|\frac{E[h^3 \cdot (\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4)]}{Var[Z_{\mathbf{M}}]^2}\right| = h^3 \cdot \left|\frac{E[\sum_{j=0}^{h-1}(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[Z_{\mathbf{M}}]^2}\right|$$
$$= h^3 \cdot \sum_{j=0}^{h-1}\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[Z_{\mathbf{M}}]^2}\right| \leqslant h^3 \cdot \sum_{j=0}^{h-1}\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|.$$

By Lemma 4.3.3,4.3.4,4.3.5 and 4.3.6, $\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$ for all $j = 0, 1, \cdots, h-1$. Therefore, the following inequality holds.

$$\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| \leqslant \mathsf{poly}(\lambda) =: q(\lambda)$$

The same holds for $\mathbf{N}$ as well.

Moreover, $Var[Z_{\mathbf{N}}] - Var[Z_{\mathbf{M}}] = \Theta\left(w^3 \cdot n \cdot m^{h-2}(\sigma^2)^h\right)$ holds by Lemma 4.3.4. Then the values $\left|Var[(Z_{1,1}^{(\mathbf{M})})_j]/(Var[Z_{\mathbf{N}}] - Var[Z_{\mathbf{M}}])\right|$ is bounded by $\mathsf{poly}(\lambda)$ for every j since $\sigma^4 < m^\ell/n^{\ell+1}$. This implies the first condition also holds.

□

Remark 4.3.2. *In the original paper [CVW18], the authors give two different choice of the distributions of $\mathbf{E}_{i,b}$; $\mathcal{D}_{\mathbb{Z},\sigma}$ with corresponding dimension in Section 11, and $\chi = \mathcal{D}_{\mathbb{Z},2\sqrt{\lambda_{LWE}}}$ with appropriate dimension in Section 5. This paper focus on $\mathcal{D}_{\mathbb{Z},\sigma}$ but the result still holds for $\chi = \mathcal{D}_{\mathbb{Z},2\sqrt{\lambda_{LWE}}}$ with slight modification.*

4.4 Cryptanalysis of BGMZ Obfuscation

In this section, we briefly review the BGMZ obfuscation and apply the statistical zeroizing attack on BGMZ obfuscation for exponentially large variance σ. Note that the security proof of BGMZ obfuscation under GGH15 zeroizing model (and underlying BPUA assumption) is independent of the parameter σ, so our attack implies that the algebraic security proof is not enough to achieve the ideal security of iO.

4.4.1 Construction of BGMZ Obfuscation

Bartusek *et al.* proposed a new candidate of iO which is provably secure in the GGH15 zeroizing model. We briefly review the construction of this scheme. For more detail, we refer to the original paper [BGMZ18].

We start with the conditions of BP they used. The authors use a dual-input binary BP's. *i.e.*, $\mathsf{inp}(i) = (\mathsf{inp}_1(i), \mathsf{inp}_2(i))$. For simplicity, they use the notation $\mathbf{x}(i) = (x_{\mathsf{inp}_1(i)}, x_{\mathsf{inp}_2(i)})$. Moreover, they employ the new parameter $\eta = \mathsf{poly}(\ell, \lambda)$ with $\eta \geqslant \ell^4$ which decides the minimum number of the BP layer for the security parameter λ and input length ℓ.

The targeted BP $\mathbf{P}$ also satisfies the following conditions.

1. All the matrices $\{\mathbf{P}_{i,\mathbf{b}}\}_{i\in[h],\mathbf{b}\in\{0,1\}^2}$ are $w \times w$ matrices.
2. $\prod_{i=1}^{h} \mathbf{P}_{i,\mathbf{x}(i)} = \mathbf{0}^{w\times w}$.
3. Each pair of input bits (j, k) is read in at least $4\ell^2$ different layers of branching program.
4. There exist layers $i_1 < i_2 < \cdots < i_\eta$ such that $\mathsf{inp}_1(i_1), \cdots, \mathsf{inp}_1(i_\eta)$ cycles η/ℓ times through $[\ell]$.

To obfuscate a branching program that does not satisfy the condition 3 or 4, one pads the identity matrices to satisfy the conditions while preserving the functionality.

Remark 4.4.1. *The original construction consider the straddling set and asymmetric level structures to prohibit invalid evaluations. The description below omitted them because our attack only exploits the valid evaluations whose results are the same regardless of them.*

Construction. BGMZ obfuscation is a probabilistic polynomial time algorithm which takes as input a BP $\mathbf{P}$ with a length h, and outputs an obfuscated program with the same functionality. We use several parameter such as $n, m, q, t := (w+1)\cdot n, \sigma, \nu, g$ in the construction. We will describe the setting for new parameters such as g, ν later.

The obfuscation procedure consists of the following steps.

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leqslant i \leqslant h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m})$, $\{\mathbf{E}_{i,\mathbf{b}} \leftarrow \chi^{t\times m}\}_{i\in[h-1],\mathbf{b}\in\{0,1\}^2}$ and $\mathbf{E}_h \leftarrow \chi^{t\times m}$ where $t := (w+1)\cdot n$.

- Sample matrices $\mathbf{B}_{i,\mathbf{b}} \in \mathbb{Z}_\nu^{g\times g}$ and invertible matrices $\mathbf{R}_i \in \mathbb{Z}_q^{(m+g)\times(m+g)}$ randomly.

- Sample matrices $\{\mathbf{S}_{i,\mathbf{b}} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}\}_{i\in[h-1],\mathbf{b}\in\{0,1\}^2}$ and a final encoding $\mathbf{D}_h$ as

$$\mathbf{D}_h \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \begin{pmatrix} \mathbf{I}^{wn\times wn} & \\ & \mathbf{0}^{n\times n} \end{pmatrix} \cdot \mathbf{A}_h + \mathbf{E}_h, \sigma),$$

and compute bookend vectors $\mathbf{v}$ and $\mathbf{w}$ as

$$\mathbf{v} = [\mathbf{v}' \cdot \mathbf{J} \cdot \mathbf{A}_0 \mid \mathbf{b}_v] \cdot \mathbf{R}_1,$$
$$\hat{\mathbf{S}}_{i,\mathbf{b}} := \begin{pmatrix} \mathbf{P}_{i,\mathbf{b}} \otimes \mathbf{S}_{i,\mathbf{b}} & \\ & \mathbf{S}_{i,\mathbf{b}} \end{pmatrix} \in \mathbb{Z}^{t \times t}$$
$$\mathbf{w}^T = \mathbf{R}_h^{-1} \cdot \begin{pmatrix} \mathbf{D}_h \cdot \mathbf{w}'^T \\ \mathbf{b}_w^T \end{pmatrix}$$

where $\mathbf{v}' \leftarrow \mathcal{D}^n_{\mathbb{Z},\sigma}$, $\mathbf{w}' \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$, $\mathbf{b}_v, \mathbf{b}_w \leftarrow U(\mathbb{Z}^k_\nu)$ and $\mathbf{J} := [\mathbf{J}'|\mathbf{I}^{n\times n}]$ with a randomly chosen matrix $\mathbf{J}' \leftarrow \{0,1\}^{n \times wn}$.

- Compute matrices

$$\mathbf{D}_{i,} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,\mathbf{b}} \cdot \mathbf{A}_i + \mathbf{E}_{i,\mathbf{b}}, \sigma) \text{ with } 1 \leqslant i \leqslant h-1,$$
$$\text{and } \mathbf{C}_{i,\mathbf{b}} = \mathbf{R}_i^{-1} \cdot \begin{pmatrix} \mathbf{D}_{i,\mathbf{b}} & \\ & \mathbf{B}_{i,\mathbf{b}} \end{pmatrix} \cdot \mathbf{R}_{i+1} \text{ with } i = 1, \cdots, h-1.$$

Evaluation. Outputs 0 if $|\mathbf{v} \cdot \prod_{i=1}^{h-1} \mathbf{C}_{i,\mathbf{x}(i)} \cdot \mathbf{w}^T| \leqslant B$. Otherwise, outputs 1.

Parameters. We first consider several security parameters. Let λ and $\lambda_{LWE} = \mathsf{poly}(\lambda)$ be security parameters depending on the obfuscation itself and the hardness of LWE satisfying following constraints, respectively. Set $n = \Omega(\lambda_{LWE} \log q)$, $\chi = \mathcal{D}_{\mathbb{Z},s}$ with $s = \Omega(\sqrt{n})$. Moreover, for the trapdoor functionality, we set $m = \Omega(t \log q)$ and $\sigma = \Omega(\sqrt{t \log q})$. In addition, they use parameters $g = 5$ and $\nu = 2^\lambda$. For correctness we set zerotest bound $B = (m \cdot \beta \cdot \sigma \cdot \sqrt{t})^{h+1} + (k \cdot \nu)^{h+1}$ and $B \cdot \omega(\mathsf{poly}(\lambda)) \leqslant q \leqslant (\sigma/\lambda_{LWE}) \cdot 2^{\lambda^{1-\epsilon}_{LWE}}$ for some fixed $\epsilon \in (0,1)$. For more detail we refer readers to the original paper [BGMZ18].

Zerotest Functionality. From the construction of obfuscation, the following equality always holds if $\mathbf{C} := \prod_{i=1}^{h-1} \mathbf{C}_{i,\mathbf{x}(i)}$ is an encoding of zero computed by honest evaluation.

$$
\begin{aligned}
&\|[\mathbf{v} \cdot \mathbf{C} \cdot \mathbf{w}^T]_q\|_\infty \\
&= \left\| \left[\mathbf{v}' \cdot \mathbf{J} \cdot \sum_{j=1}^{h} ((\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,\mathbf{x}(i)}) \cdot \mathbf{E}_{j,\mathbf{x}(j)} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,\mathbf{x}(k)} \cdot \mathbf{w}'^T + \mathbf{b}_v \cdot \prod_{i=1}^{h-1} \mathbf{B}_{i,\mathbf{x}(i)} \cdot \mathbf{b}_w^T \right]_q \right\|_\infty \\
&\leqslant \left\| \mathbf{v}' \cdot \mathbf{J} \cdot \sum_{j=1}^{h} ((\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,\mathbf{x}(i)}) \cdot \mathbf{E}_{j,\mathbf{x}(j)} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,\mathbf{x}(k)} \cdot \mathbf{w}'^T + \mathbf{b}_v \cdot \prod_{i=1}^{h-1} \mathbf{B}_{i,\mathbf{x}(i)} \cdot \mathbf{b}_w^T \right\|_\infty \\
&\leqslant \sigma^2 \cdot m^2 \cdot (m \cdot \beta \cdot \sigma \cdot \sqrt{t})^{h-1} + (k \cdot \nu)^{h+1}
\end{aligned}
$$

Since $\|[\mathbf{v}\cdot\mathbf{C}\cdot\mathbf{w}^T]_q\|_\infty$ is bounded by $\sigma^2\cdot m^2\cdot(m\cdot\beta\cdot\sigma\cdot\sqrt{t})^{h-1}+(k\cdot\nu)^{h+1} \leqslant B$ for all but negligible probability. Moreover, if $\prod_{i=1}^{h} \mathbf{P}_{i,\mathbf{x}(i)}$ is a nonzero matrix, then $\prod_{i=1}^{h} \hat{\mathbf{S}}_{i,\mathbf{x}(i)}$ is also nonzero matrix. Thus, $\|[\mathbf{v} \cdot \mathbf{C} \cdot \mathbf{w}^T]_q\|_\infty$ is larger than B with overwhelming probability because of $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m})$.

4.4.2 Cryptanalysis of BGMZ Obfuscation

In this section, we analyze the conditions for the statistical zeroizing attack on the BGMZ obfuscation when we assume $\sigma \geqslant \nu = 2^\lambda$. (More precisely, the same result holds when $\sigma^2 \geqslant \nu^2 g/12m$.). As in Section 4.3.2, the notation written in the capital italic words are regarded as the random matrix whose entry follows a distribution that corresponds to the distribution of entry of the bold-written matrix.

The targeted BPs are $\mathbf{M} = \{\mathbf{M}_{i,\mathbf{b}}\}_{i\in[h],\mathbf{b}\in\{0,1\}^2}$ and $\mathbf{N} = \{\mathbf{N}_{i,\mathbf{b}}\}_{i\in[h],\mathbf{b}\in\{0,1\}^2}$

such that

$$\mathbf{M}_{i,\mathbf{b}} = \mathbf{0}^{w\times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,\mathbf{b}} = \begin{cases} \mathbf{I}^{w\times w} & \text{if } i = 1 \\ \mathbf{0}^{w\times w} & \text{otherwise} \end{cases}.$$

Note that two branching programs always output zero. Now we suppose that we have polynomially many samples from the one of two distributions $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$, where $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ are the distributions of the evaluations of obfuscations of $\mathbf{M}$ and $\mathbf{N}$.

Then our purpose is to distinguish whether the samples come from $\mathcal{D}_{\mathbf{M}}$ or $\mathcal{D}_{\mathbf{N}}$ by Proposition 4.2.1. We obtain random matrices $S^{(\mathbf{P})}_{i,\mathbf{b}}$, $E^{(\mathbf{P})}_{i,\mathbf{b}}$, $D^{(\mathbf{P})}_{i,\mathbf{b}}$ and $C^{(\mathbf{P})}_{i,\mathbf{b}}$ as in the construction of BGMZ obfuscation for branching programs $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$. Thus, it suffices to prove the following proposition.

Proposition 4.4.1. *Let λ be a security parameter and σ the Gaussian variance parameter satisfying $\sigma^2 \geqslant \nu^2 g/12m$ for parameters m, ν and g of BGMZ obfuscation. Then, there are two functionally equivalent branching programs $\mathbf{M}$ and $\mathbf{N}$ satisfying the following statement: let $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$ be random variables satisfying*

$$Z_{\mathbf{M}} = \left[v \cdot \prod_{i=1}^{h-1} C^{(\mathbf{M})}_{i,\mathbf{x}(i)} \cdot w^T \right]_q \text{ and } Z_{\mathbf{N}} = \left[v \cdot \prod_{i=1}^{h-1} C^{(\mathbf{N})}_{i,\mathbf{x}(i)} \cdot w^T \right]_q.$$

where every random matrix is defined as the above. Let $\mu_{\mathbf{M}}$ and $\mu_{\mathbf{N}}$, $\sigma^2_{\mathbf{M}}$ and $\sigma^2_{\mathbf{N}}$ be mean and variance of the random variables of $Z_{\mathbf{M}}$ and $Z_{\mathbf{N}}$, respectively. Then, it holds that

$$\left| \frac{\max(\sigma^2_{\mathbf{N}}, \sigma^2_{\mathbf{M}})}{\sigma^2_{\mathbf{N}} - \sigma^2_{\mathbf{M}}} \right| \leqslant p, \ \left| \frac{E[(Z_{\mathbf{N}} - \mu_{\mathbf{N}})^4]}{\sigma^4_{\mathbf{N}}} \right| \leqslant q, \text{ and } \left| \frac{E[(Z_{\mathbf{M}} - \mu_{\mathbf{M}})^4]}{\sigma^4_{\mathbf{M}}} \right| \leqslant q.$$

for some $p, q = \mathsf{poly}(\lambda)$ *under Assumption 1.*

Note that Assumption 1 (for BGMZ obfuscation) is also needed to verify the proposition. With the honest evaluation $\left[\mathbf{v} \cdot \prod_{i=1}^{h-1} \mathbf{C}_{i,\mathbf{x}(i)} \cdot \mathbf{w}^T\right]_q$ of the BGMZ obfuscation, we obtain the integer of the form

$$\mathbf{v}' \cdot \mathbf{J} \sum_{j=1}^{h} ((\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,\mathbf{x}(i)}) \mathbf{E}_{j,\mathbf{x}(j)} \prod_{k=j+1}^{h} \mathbf{D}_{k,\mathbf{x}(k)} \cdot \mathbf{w}'^T + \mathbf{b}_v \cdot \prod_{i=1}^{h-1} \mathbf{B}_{i,\mathbf{x}(i)} \cdot \mathbf{b}_w^T$$

which does not contain the term including trapdoor matrices $\mathbf{A}_i$'s. Thus, similarly to the CVW obfuscation case, we need to analyze the statistical properties of the random vectors $v'^{(\mathbf{P})}$, $w'^{(\mathbf{P})}$, $b_v^{(\mathbf{P})}$, $b_w^{(\mathbf{P})}$ and random matrices $\hat{S}_{i,\mathbf{b}}^{(\mathbf{P})}$, $E_{i,\mathbf{b}}^{(\mathbf{P})}$, $D_{i,\mathbf{b}}^{(\mathbf{P})}$ and their products to prove the statistical properties including the variance in Proposition 4.4.1.

The proof of Proposition 4.4.1 is based on the following lemmas and placed in the concluding part of this section. All proofs of these lemmas are in Appendix 6.2.8. Note that most lemmas in this section also hold under Assumption 1 as the section 4.3.2, so we omit repeated *under Assumption 1* in statements. Notations c_0, c, and d are similarly defined as Section 4.3.

For $j = 0, 1, \cdots, h-1$, let $(Z^{(\mathbf{M})})_j$ be a random variable of the form

$$v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot \prod_{i=1}^{j} \hat{S}_{i,\mathbf{x}(i)}^{(\mathbf{M})} \cdot E_{j+1,\mathbf{x}(j+1)}^{(\mathbf{M})} \cdot \prod_{k=j+2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})} \cdot w'^{(\mathbf{M})^T},$$

and for $j = h$, $(Z^{(\mathbf{M})})_h$ a random variable of the form

$$b_v^{(\mathbf{M})} \cdot \prod_{i=1}^{h-1} B_{i,\mathbf{x}(i)}^{(\mathbf{M})} \cdot b_w^{(\mathbf{M})^T}.$$

We similarly define $(Z^{(\mathbf{N})})_j$ for $j = 0, 1, \cdots, h$, and $Z_{\mathbf{P}} = \sum_{i=0}^{h} (Z^{(\mathbf{P})})_j$ for

$\mathbf{P} = \mathbf{M}$ and $\mathbf{N}$.

Lemma 4.4.1. *$E[(Z^{(\mathbf{M})})_j] = E[(Z^{(\mathbf{N})})_j] = 0$ for all $j = 0, 1, \cdots, h$.*

Lemma 4.4.2. *$E[(Z^{(\mathbf{M})})_{\mu_1} \cdot (Z^{(\mathbf{M})})_{\mu_2}] = E[(Z^{(\mathbf{N})})_{\mu_1} \cdot (Z^{(\mathbf{N})})_{\mu_2}] = 0$ for $\mu_1 \neq \mu_2$.*

Lemma 4.4.3 ($j = 0$)**.** *It holds that*

$$Var[(Z^{(\mathbf{M})})_0] = Var[(Z^{(\mathbf{N})})_0] = \Theta\left(wn \cdot m^h \cdot (\sigma^2)^{h+1} \cdot s^2\right),$$
$$\left|\frac{E[(Z^{(\mathbf{M})})_0^4]}{Var[(Z^{(\mathbf{M})})_0]^2}\right|, \left|\frac{E[(Z^{(\mathbf{N})})_0^4]}{Var[(Z^{(\mathbf{N})})_0]^2}\right| \leqslant 108c_0(w+1)^2 \cdot n^2m^4 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.4.4 ($j = 1$)**.** *It holds that*

$$Var[(Z^{(\mathbf{M})})_1] = \Theta\left(n^2m^{h-1} \cdot (\sigma^2)^{h+1} \cdot s^2\right),$$
$$Var[(Z^{(\mathbf{N})})_1] = \Theta\left(wn^3m^{h-1} \cdot (\sigma^2)^{h+1} \cdot s^2\right) + Var[(Z^{(\mathbf{M})})_1]$$

Moreover, it holds that

$$\left|\frac{E[(Z^{(\mathbf{M})})_1^4]}{Var[(Z^{(\mathbf{M})})_1]^2}\right| \leqslant 81c_0 \cdot n^4m^4 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda),$$
$$\left|\frac{E[(Z^{(\mathbf{N})})_1^4]}{Var[(Z^{(\mathbf{N})})_1]^2}\right| \leqslant 324c_0(w+1)^2 \cdot n^6m^4 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.4.5 ($2 \leqslant j \leqslant h-1$)**.** *It holds that*

$$Var[(Z^{(\mathbf{M})})_j] = Var[(Z^{(\mathbf{N})})_j] = \Theta\left(n^{j+1}m^{h-j} \cdot (\sigma^2)^{h+1} \cdot s^2\right).$$

Moreover, it holds that

$$\left|\frac{E[(Z^{(\mathbf{M})})_j^4]}{Var[(Z^{(\mathbf{M})})_j]^2}\right|, \left|\frac{E[(Z^{(\mathbf{N})})_j^4]}{Var[(Z^{(\mathbf{N})})_j]^2}\right| \leqslant 81c_0 \cdot n^4 m^4 \left(1+\frac{2}{n}\right)^{j-1} \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Lemma 4.4.6 ($j = h$)**.** *It holds that*

$$Var[(Z^{(\mathbf{M})})_h] = Var[(Z^{(\mathbf{N})})_h] = g^h \cdot \left\{\frac{1}{12} \cdot \nu(\nu+2)\right\}^{h+1}.$$

Moreover, it holds that

$$E[(Z^{(\mathbf{M})})_h^4], E[(Z^{(\mathbf{N})})_h^4] \leqslant 27 \cdot (g^2)^4 \cdot \{g(g+2)\}^{h-2} \cdot \left\{\frac{1}{12} \cdot \nu(\nu+2)\right\}^{2(h+1)}.$$

Now we give a proof of the proposition 4.4.1 using the above lemmas.

of Proposition 4.4.1. Choose BPs $\mathbf{M}$ and $\mathbf{N}$ as given in the first page of this section. They have the same functionality and length.

Note that elements $(Z^{(\mathbf{M})})_j$ in the above Lemmas are of the form

$$(Z^{(\mathbf{M})})_j = v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot \prod_{i=1}^{j} \hat{S}^{(\mathbf{M})}_{i,\mathbf{x}(i)} \cdot E^{(\mathbf{M})}_{j+1,\mathbf{x}(j+1)} \cdot \prod_{k=j+2}^{h} D^{(\mathbf{M})}_{k,\mathbf{x}(k)} \cdot w'^{(\mathbf{M})^T} \text{ for } j < h$$

$$(Z^{(\mathbf{M})})_h = b_v^{(\mathbf{M})} \cdot \prod_{i=1}^{h-1} B^{(\mathbf{M})}_{i,\mathbf{x}(i)} \cdot b_w^{(\mathbf{M})^T}$$

Let $Z_{\mathbf{M}}$ be the summation of $(Z^{(\mathbf{M})})_j$ for $j \in \{0, 1, \cdots, h\}$. From Lemma 4.4.2, we have

$$Var[Z_{\mathbf{M}}] = E\left[(\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i)^2\right] = E\left[\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^2\right] = \sum_{i=0}^{h} Var[(Z^{(\mathbf{M})})_i],$$

$$E[Z_{\mathbf{M}}^4] = E\left[(\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i)^4\right] \leqslant E\left[(h+1)^3 \cdot (\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4)\right].$$

After dividing both sides by $Var[Z_{\mathbf{M}}]^2$, we obtain the following inequality

$$\begin{aligned}\left|\frac{E[Z_{\mathbf{M}}^4]}{Var[Z_{\mathbf{M}}]^2}\right| &\leqslant \left|\frac{E[(h+1)^3 \cdot (\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4)]}{Var[Z_{\mathbf{M}}]^2}\right| = (h+1)^3 \cdot \left|\frac{E[\sum_{i=0}^{h}(Z^{(\mathbf{M})})_i^4]}{Var[Z_{\mathbf{M}}]^2}\right| \\ &= (h+1)^3 \cdot \sum_{i=0}^{h}\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[Z_{\mathbf{M}}]^2}\right| \\ &\leqslant (h+1)^3 \cdot \left(\sum_{i=0}^{h-1}\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[(Z^{(\mathbf{M})})_i]^2}\right| + \left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[Z_{\mathbf{M}}]^2}\right|\right)\end{aligned}$$

By Lemma 4.4.3,4.4.4, 4.4.5 and 4.4.6, $\left|\frac{E[(Z^{(\mathbf{M})})_i^4]}{Var[(Z^{(\mathbf{M})})_i]^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$ for all $i = 0, 1, \cdots, h-1$ regardless of $\mathbf{P} = \mathbf{M}$ or $\mathbf{P} = \mathbf{N}$. Since $\sigma^2 \geqslant \nu^2 g/12m$, we obtain the following upper bound.

$$\begin{aligned}\left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[Z_{\mathbf{M}}]^2}\right| &\leqslant \left|\frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[(Z^{(\mathbf{M})})_0]^2}\right| \\ &= O\left((g^2)^4 \cdot \left(\frac{g(g+2)}{m^2}\right)^{h-2} \cdot \left(\frac{\nu(\nu+2)}{12\sigma^2}\right)^{h+1}\right) \\ &= \mathsf{poly}(\lambda)\end{aligned}$$

Thus the kurtosis is bounded by polynomial of security parameter λ.

Moreover, by the definition of $Z_{\mathbf{N}}$ and $Z_{\mathbf{M}}$ and lemmas, we obtain the equality $|\sigma_{\mathbf{N}}^2 - \sigma_{\mathbf{M}}^2| = \Theta\left(wn^3m^{h-1} \cdot (\sigma^2)^{h+1} \cdot s^2\right)$. Using lemmas, $\left|\frac{\max(\sigma_{\mathbf{N}}^2, \sigma_{\mathbf{M}}^2)}{\sigma_{\mathbf{N}}^2 - \sigma_{\mathbf{M}}^2}\right|$ is bounded by $\mathsf{poly}(\lambda)$. □

Chapter 5

Conclusions

In this paper, we proposed mathematical analyses of branching program iO based on GGH13 and GGH15 multilinear maps.

First, in case of indistinguishability obfuscation candidates based on GGH13, we showed that if NTRU-solver exists, then the all known iO candidates over GGH13 do not obtain the desired security. In other words, there exists two functionally equivalent branching programs such that their obfuscated programs are distinguishable in polynomial time.

Second, we proposed a new cryptanalysis of iO based on GGH15, called the statistical zeroizing attack. Unlike the previous works, we proposed the first statistical attack to iO schemes based on GGH15. As the results, we broke the CVW obfuscation for suggested parameters, and showed that algebraic security model assumed by BGMZ obfuscation is insufficient to achieve ultimate security model of iO. Indeed, we showed that the statistical zeroizing attack is lying outside of the algebraic security model by suggesting some parameters that holds the algebraic security model, but are insecure under the attack.

Chapter 6

Appendix

6.1 Appendix of Chapter 3

6.1.1 Extended Attackable Model

In this section we introduce an extended model of attackable BP obfuscation by our attack. The extended attackable BP obfuscation is modified in the randomization step to embraces the obfuscation in [BR14]. The definition of extended attackable conditions for randomization is as follows, which is similar to Definition 3.3.1:

Definition 6.1.1 (Extended Attackable Conditions for Randomization)**.** *For a branching program* $P = \left\{\mathbf{M}_{i,\mathbf{b}} \in \mathbb{Z}^{d_i \times d_{i+1}}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}$*, the extended attackable randomized branching program is the set*

$$\begin{aligned} Rand(P) &= \left\{\mathbf{R}_{i,\mathbf{b}}, \mathbf{R}'_{i,\mathbf{b}} \in \mathbb{Z}^{d_i \times d_{i+1}}\right\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w} \\ &\cup \left\{\mathbf{R}_{\mathbf{S}}, \mathbf{R}'_{\mathbf{S}} \in \mathbb{Z}^{d_0 \times d_1}, \mathbf{R}_{\mathbf{T}}, \mathbf{R}'_{\mathbf{T}} \in \mathbb{Z}^{d_{\ell+1} \times d_{\ell+2}}\right\} \\ &\cup \left\{\mathsf{aux}_{J,\mathbf{b}}, \mathsf{aux}'_{J,\mathbf{b}}\right\}_{J\subset[N],\mathbf{b}\in\{0,1\}^{w\times|J|}} \end{aligned}$$

satisfying the following properties, where $d_0, d_{\ell+2}, e_i$'s are integers.
1. There exist matrices $\mathbf{S}_0, \mathbf{S}'_0 \in \mathbb{Z}^{d_0 \times d_1}, \mathbf{T}_0, \mathbf{T}'_0 \in \mathbb{Z}^{d_\ell \times d_{\ell+1}}$ and scalars $\alpha_{\mathbf{S}}, \alpha'_{\mathbf{S}}, \alpha_{\mathbf{T}}, \alpha'_{\mathbf{T}}, \{\alpha_{i,\mathbf{b}}, \alpha'_{i,\mathbf{b}}\}_{i\in[\ell],\mathbf{b}\in\{0,1\}^w}$ such that the following equations hold for all $\{\mathbf{b}_i \in \{0,1\}^w\}_{i\in[\ell]}$:

$$\mathbf{R}_S \cdot \prod_{i=1}^{\ell} \mathbf{R}_{i,\mathbf{b}_i} \cdot \mathbf{R}_T = \alpha_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha_{i,\mathbf{b}_i} \cdot \alpha_{\mathbf{T}} \cdot \left(\mathbf{S}_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{T}_0 \right),$$
$$\mathbf{R}'_S \cdot \prod_{i=1}^{\ell} \mathbf{R}'_{i,\mathbf{b}_i} \cdot \mathbf{R}'_T = \alpha'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \alpha'_{i,\mathbf{b}_i} \cdot \alpha'_{\mathbf{T}} \cdot \left(\mathbf{S}'_0 \cdot \prod_{i=1}^{\ell} \mathbf{M}'_{i,\mathbf{b}_i} \cdot \mathbf{T}'_0 \right).$$

2. The evaluation of randomized program is done by checking whether the fixed entries of

$$RP(\mathbf{x}) = \prod_{J\subset[N]} \mathsf{aux}_{J,\mathbf{x}|_J} \cdot \mathbf{R}_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{R}_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{R}_{\mathbf{T}} - \prod_{J\subset[N]} \mathsf{aux}'_{J,\mathbf{x}|_J} \cdot \mathbf{R}'_{\mathbf{S}} \cdot \prod_{i=1}^{\ell} \mathbf{R}'_{i,\mathbf{x}_{\mathsf{inp}(i)}} \cdot \mathbf{R}'_{\mathbf{T}}$$

is zero or not. Especially, there are two integers u, v such that $P(\mathbf{x}) = 0 \Rightarrow RP(\mathbf{x})[u, v] = 0$.

After randomizing matrices, we encode every entries and scalars of $Rand(P)$ separately by GGH13 multilinear map with respect to the level corresponding to the first index of elements. We denote $\mathsf{enc}(\mathsf{aux}_{J,\mathbf{a}})$ by $\widetilde{\mathsf{aux}}_{J,\mathbf{a}}$ for each $J \subset [N]$ and $\mathbf{a} \in \{0,1\}^{w\times|J|}$.

We note that aux's were not discussed in the main body of our paper. However, our program converting technique is applied with small modification for auxiliary scalars as well. More precisely, for each $\widetilde{\mathsf{aux}}_{J,\mathbf{a}}, \widetilde{\mathsf{aux}}_{J,\mathbf{b}}$, we compute $\mathbf{h} = \widetilde{\mathsf{aux}}_{J,\mathbf{a}}/\widetilde{\mathsf{aux}}_{J,\mathbf{b}}$ and solve the NTRU problem for the instance $\mathbf{h}$. Then we obtain $\mathbf{c}_J \cdot (\mathsf{aux}_{J,\mathbf{a}} + \mathbf{r}_{\mathbf{a}} \cdot \mathbf{g})$ for small $\mathbf{c}_J$. For an auxiliary scalar $\widetilde{\mathsf{aux}}_{J,\mathbf{c}}$ corresponding to J, we compute $\mathbf{c}_J \cdot (\mathsf{aux}_{J,\mathbf{c}} + \mathbf{r}_{\mathbf{c}} \cdot \mathbf{g}) =$

$\mathbf{c}_J \cdot (\mathsf{aux}_{J,\mathbf{a}} + \mathbf{r}_{\mathbf{a}} \cdot \mathbf{g}) \cdot \widetilde{\mathsf{aux}}_{J,\mathbf{c}}/\widetilde{\mathsf{aux}}_{J,\mathbf{a}}$. We can recover dummy auxiliaries as well.

From this calculation, $\mathcal{R}$ program is obtained for extended model. the other step such as recovering the ideal $\langle \mathbf{g} \rangle$ and the matrix zeroizing attack work correctly as well.

6.1.2 Examples of Matrix Zeroizing Attack

Obfuscation in [PST14].

In this section, we prove that obfuscation in [PST14] cannot be iO for general-purpose. This scheme is characterized by several special randomizations; converting to merged branching program which consists of permutation matrices, and choose the right bookend vector $\mathbf{T} = \mathbf{e}_1$ and no left bookend vector, and then choose identity Kilian matrix $\mathbf{K}_0 = \mathbf{I}$ at the first left position. It implies that, by Proposition 3.4.2, the evaluation of the program is of the form:

$$\prod_{i=1}^{\ell} \mathbf{D}_{i,\mathbf{b}_i} \cdot \mathbf{D}_{\mathbf{T}} = \rho_{\mathbf{T}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i} \cdot \prod_{i=1}^{\ell} \mathbf{M}_{i,\mathbf{b}_i} \cdot \mathbf{e}_1 = \rho_{\mathbf{T}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i} \cdot \mathbf{e}_k \ (\mathrm{mod} \langle \mathbf{g} \rangle),$$

where k is an integer computed by $\mathbf{M}$'s. Therefore, we can compute $\rho_{\mathbf{T}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i}$ from the computed value. As a next step, we recover ratios of scalar bundlings $\rho_{j,\mathbf{b}_j}/\rho_{j,\mathbf{b}'_j}$ for $\mathbf{b}, \mathbf{b}'$ which satisfies $\mathbf{b}_i = \mathbf{b}'_i$ for all $i \in [\ell]$ except j by computing the ratio $\rho_{\mathbf{T}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}_i}/\rho_{\mathbf{T}} \cdot \prod_{i=1}^{\ell} \rho_{i,\mathbf{b}'_i}$. Finally, we can run the matrix zeroizing attack.

Obfuscation in [BMSZ16].

Badrinarayanan *et al.* suggest a construction for obfuscation based on branching program, especially for *evasive functions* [BMSZ16].*. In this section, we prove that obfuscation of Badrinarayanan *et al.* cannot be a general-purpose iO. This construction is for low-rank branching program, thus it do not have dummy matrices and also does not apply higher dimension embeddings.

The original method for their construction is in the bookend; the authors use no bookend matrices and use special form of Kilian randomization at the first and last matrices. The first and last Kilian matrices are given as follows:

$$\mathbf{K}_0 = diag(\beta_1, \cdots, \beta_{d_1}), \mathbf{K}_{\ell+1}^{-1} = diag(\gamma_1, \cdots, \gamma_{d_{\ell+1}}),$$

where β_u, γ_v are randomly chosen scalars.

To evaluate the obfuscated program, we see $\left(\prod_{i=1}^{\ell} \widetilde{M}_{i,\mathbf{b}_i}\right)[u,v]$ for some u, v. This is corresponding to the following value, which is computed by Proposition 3.4.2,

$$\left(\prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}_i}\right)[u,v] = \beta_u \cdot \gamma_v \cdot \prod_{i\in[\ell]} \rho_{i,\mathbf{b}_i} \cdot \left(\prod_{i\in[\ell]} \mathbf{M}_{i,\mathbf{b}_i}\right)[u,v] \pmod{\langle \mathbf{g} \rangle}$$

since $\mathbf{S}_0, \mathbf{T}_0$ are exactly $\mathbf{K}_0, \mathbf{K}_{\ell+1}^{-1}$. We then can recover the ratio of scalar bundlings by computing $\prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}_i}[u,v] / \prod_{i\in[\ell]} \mathbf{D}_{i,\mathbf{b}'_i}[u,v]$ for $\mathbf{b}, \mathbf{b}'$ which satisfies $\mathbf{b}_i = \mathbf{b}'_i$ for all $i \in [\ell]$ except j. Since we computed ratios of scalar

*We remark that the construction of [BMSZ16] is similar to the construction of [SZ14], which is used as a foundation of recent implementation 5Gen [LMA+16] and our attack is also applied to [SZ14] in the same manner.

bundlings $\rho_{j,\mathbf{b}_j}/\rho_{j,\mathbf{b}'_j}$, we can run the matrix zeroizing attack.

6.1.3 Examples of Linear Relationally Inequivalent BPs

We exhibit two examples of two functionally equivalent but linear relationally inequivalent branching programs here. This examples also certify Proposition 3.3.2. The first simple example from nondeterministic finite automata is read-once BPs, and the second example comes from Barrington's theorem and thus input-unpartitionable.

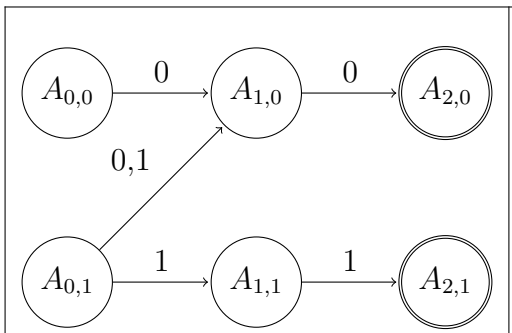
6.1.4 Read-once BPs from NFA

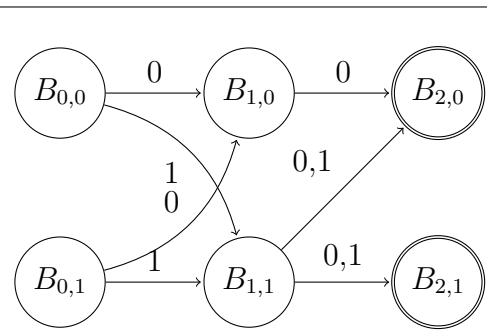
Two read-once BPs in Table 3.1 are from non-deterministic finite automata and linear relationally inequivalent.

These two BPs are the point function which output 1 only for input 01, but they are linear relationally inequivalent. For example,

$$\mathbf{M}_{0,1} \cdot \mathbf{M}_{1,0} - \mathbf{M}_{0,1} \cdot \mathbf{M}_{1,1} \neq \mathbf{0},$$
$$\mathbf{N}_{0,1} \cdot \mathbf{N}_{1,0} - \mathbf{N}_{0,1} \cdot \mathbf{N}_{1,1} = \mathbf{0}.$$

We note that the matrix $\mathbf{M}_{i,b}$ is the adjacent matrix between $\{A_{i,c}\}_{c\in\{0,1\}}$ and $\{A_{i+1,c}\}_{c\in\{0,1\}}$, and $\mathbf{N}$'s are defined similarly.





$$\mathbf{M}_{0,0} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \ \mathbf{M}_{1,0} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\mathbf{M}_{0,1} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \ \mathbf{M}_{1,1} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

$$\mathbf{N}_{0,0} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \ \mathbf{N}_{1,0} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

$$\mathbf{N}_{0,1} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \ \mathbf{N}_{1,1} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

Table 6.1: BPs from NFA

6.1.5 Input-unpartitionable BPs from Barrington's Theorem

In the case of Barrington's theorem, the linear relationally inequivalent matrix BPs are more complex. We consider the following two functionally equivalent circuits:

$$C_0 = (X_1 \wedge X_2) \wedge (\neg X_1 \wedge X_3),$$
$$C_1 = (\neg X_1 \wedge X_2) \wedge (X_1 \wedge X_3).$$

We transform two circuits into the following BPs by Barrington theorem as follow[†]:

$P_{C_0} =$	0:	α_ρ	β_ρ	α_ρ^{-1}	β_ρ^{-1}	e	β_δ	e	β_δ^{-1}	$\cdots$
	1:	e	e	e	e	α_δ	e	α_δ^{-1}	e	$\cdots$
$P_{C_1} =$	0:	e	β_ρ	e	β_ρ^{-1}	α_δ	β_δ	α_δ^{-1}	β_δ^{-1}	$\cdots$
	1:	α_ρ	e	α_ρ^{-1}	e	e	e	e	e	$\cdots$
input bits		1	2	1	2	1	3	1	3	$\cdots$

where τ_σ denotes $\sigma\tau\sigma^{-1}$ for permutations $\tau, \sigma \in S_5$. In the matrix representation, the permutations $\alpha, \beta, \gamma, \rho, \delta$ are of the form

$$\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \beta = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \gamma = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\rho = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \delta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

We note that two functionally equivalent branching programs P_{C_0} and P_{C_1} are clearly input-unpartitionable. Now if we consider two (invalid) inputs $\mathbf{x} = 0110110111111111$ and $\mathbf{y} = 1111101011111111$. These yield, for example, $P_{C_0}(\mathbf{x}) = \alpha_\rho \cdot e \cdot e \cdot \beta_\rho^{-1} \cdot \alpha_\delta \cdot e \cdot e \cdot e \cdot \cdots = \alpha_\rho \cdot \beta_\rho^{-1} \cdot \alpha_\delta = \beta$.

[†]Barrington theorem can be implemented in various ways, but we only consider the first description in [Bar86]. This description also can be found in [ADGM17].

The terms in the right $\cdots$ are canceled. Then the equation

$$P_{C_0}(\mathbf{x}) - P_{C_0}(\mathbf{y}) = 0,$$
$$P_{C_1}(\mathbf{x}) - P_{C_1}(\mathbf{y}) \neq 0$$

hold. Thus two branching programs P_{C_0} and P_{C_1} are functionally equivalent but linear relationally inequivalent.

6.2 Appendix of Chapter 5

6.2.1 Simple GGH15 obfuscation

We briefly describe the construction of single input BP obfuscation based GGH15 without safeguard.

For an index to input function $\mathsf{inp} : [h] \to [\ell]$, let

$$\mathbf{P} = \left\{\mathsf{inp}, \{\mathbf{P}_{i,b} \in \{0,1\}^{w \times w}\}_{i \in [h], b \in \{0,1\}}, \mathcal{P}_0 = \mathbf{0}^{w \times w}, \mathcal{P}_1 = \mathbb{Z}^{w \times w} \backslash \mathcal{P}_0\right\}$$

be a single input BP.

For parameters $w, m, q, B \in \mathbb{N}$ and $\sigma \in \mathbb{R}^+$, the BP obfuscation based GGH15 consists of the matrices and input function, namely

$$\mathcal{O}(\mathbf{P}) = \left\{\mathsf{inp}, \mathbf{A}_0, \{\mathbf{D}_{i,b} \in \mathbb{Z}^{m \times m}\}_{i \in [h], b \in \{0,1\}}\right\}.$$

In this case, the matrix $\mathbf{T}$ in the abstract model is the identity matrix and $\mathbf{S} = \mathbf{A}_0$. The output of the obfuscation at $\mathbf{x}$ is computed as follows: compute the matrix $\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}} \bmod q$ and compare its $\| \cdot \|_\infty$ to a zerotest bound B. If it is less than B, outputs zero. Otherwise, outputs 1.

The algorithm to construct an obfuscated program $\mathcal{O}(\mathbf{P})$ proceeds as follows:

- Sample matrices $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^w, 1^m, q)$ for $i = 0, 1, \cdots, h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{w \times m})$ and $\mathbf{E}_{i,b} \leftarrow \chi^{w \times m}$ where χ is a distribution related to the hardness of LWE problem.

- By using the trapdoor τ_i, sample matrices

  $$\mathbf{D}_{i,b} \in \mathbb{Z}^{m \times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \mathbf{P}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ with } 1 \leqslant i \leqslant h.$$

- Output matrices $\{\mathbf{A}_0, \{\mathbf{D}_{i,b} \in \mathbb{Z}^{m \times m}\}_{i \in [h], b \in \{0,1\}}\}$.

Then, we observe the product $\mathcal{O}(\mathbf{P})(\mathbf{x}) = [\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}}]_q$ is equal to

$$\prod_{i=1}^{h} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \cdot \mathbf{A}_h + \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \right) \cdot \mathbf{E}_{j,x_{\mathsf{inp}(j)}} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(k)}} \right)$$

over $\mathbb{Z}_q$. If $\prod_{i=1}^{h} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} = \mathbf{0}^{w \times w}$, then $\mathcal{O}(\mathbf{P})(\mathbf{x})$ can be regarded as a summation of matrices over integers instead of $\mathbb{Z}_q$ under the certain choice of parameters as follows

$$\mathcal{O}(\mathbf{P})(\mathbf{x}) = \left[\mathbf{A}_0 \cdot \prod_{i=1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(i)}} \right]_q = \sum_{j=1}^{h} \left(\left(\prod_{i=1}^{j-1} \mathbf{P}_{i,x_{\mathsf{inp}(i)}} \right) \cdot \mathbf{E}_{j,x_{\mathsf{inp}(j)}} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{i,x_{\mathsf{inp}(k)}} \right)$$

since the infinity norm of the above matrix is less than $B \ll q$. Note that the evaluation values only rely on the matrices $\mathbf{P}_{i,b}$, $\mathbf{E}_{i,b}$ and $\mathbf{D}_{i,b}$. Thus, the evaluation result depends on the message matrices $\mathbf{P}_{i,b}$.

Suppose that we have two functionally equivalent BPs $\mathbf{M} = \{\mathbf{M}_{i,b}\}_{i \in [h], b \in \{0,1\}}$

and $\mathbf{N} = \{\mathbf{N}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ satisfies

$$\mathbf{M}_{i,b} = \mathbf{0}^{w\times w} \text{ for all } i, b \text{ and } \mathbf{N}_{i,b} = \begin{cases} \mathbf{I}^{w\times w} & \text{if } i = 1 \\ \mathbf{0}^{w\times w} & \text{otherwise} \end{cases},$$

and an obfuscated program $\mathcal{O}(\mathbf{P})$. The goal of adversary is to determine whether $\mathbf{P}$ is $\mathbf{M}$ or not. For all $\mathbf{x} \in \{0,1\}^\ell$, the evaluation of the obfuscation is of the form

$$\mathcal{O}(\mathbf{M})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} \text{ and}$$
$$\mathcal{O}(\mathbf{N})(\mathbf{x}) = \mathbf{E}_{1,x_{\mathsf{inp}(1)}} \cdot \prod_{k=2}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}} + \mathbf{I} \cdot \mathbf{E}_{2,x_{\mathsf{inp}(2)}} \cdot \prod_{k=3}^{h} \mathbf{D}_{k,x_{\mathsf{inp}(k)}}.$$

Note that they correspond to the distributions $\mathcal{D}_{\mathbf{M}}$ and $\mathcal{D}_{\mathbf{N}}$ for a fixed vector $\mathbf{x}$. These equations show the difference of two distributions in this case.

6.2.2 Modified CVW Obfuscation

We give a modification of CVW obfuscation, which can obfuscate the permutation matrix branching programs. This modification is, as far as we know, robust against all existing attacks. We first describe the transformation of branching programs. Then, we describe the modification of CVW obfuscation.

6.2.3 Transformation of Branching Programs

We first introduce the transformation from single-input permutation matrix branching programs to *Type I* BP. This transformation is applicable to BPs which outputs 0 when the product of BP matrices is the identity matrix. The output of transformation is a new branching program that outputs 0 when the product of BP matrices is the zero matrix. Through this transformation, the width of branching program is doubled. Note that this is adapted version of [CVW18, Claim 6.2].

We are given a branching program with input size ℓ

$$\mathbf{P} = \left\{ \{\mathbf{P}_{i,b} \in \{0,1\}^{w \times w}\}_{i \in [h], b \in \{0,1\}}, \mathsf{inp} : [h] \to [\ell] \right\}$$

where the evaluation of $\mathbf{P}$ at $\mathbf{x} \in \{0,1\}^{\ell}$ is computed by

$$\mathbf{P}(\mathbf{x}) = \begin{cases} 0 & \text{if } \prod_{i=1}^{h} \mathbf{P}_{i,(x_{\mathsf{inp}(i)})} = \mathbf{I}_w \\ 1 & \text{otherwise} \end{cases}$$

Then the transformation is done by changing branching program matrices as

$$\mathbf{P}' = \left\{ \left\{ \mathbf{P}'_{i,b} = \begin{pmatrix} \mathbf{P}_{i,b} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_w \end{pmatrix} \in \{0,1\}^{2w \times 2w} \right\}_{i \in [h], b \in \{0,1\}} , \mathsf{inp} : [h] \to [\ell] \right\}$$

and the evaluation is similar but uses new vectors $\mathbf{v}' = (\mathbf{v}| - \mathbf{v})$ and $\mathbf{w}' = (\mathbf{w}|\mathbf{w})$ for $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^w$:

$$\mathbf{P}'(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{v}' \cdot \prod_{i=1}^{h} \mathbf{P}'_{i,(x_{\mathsf{inp}(i)})} \cdot \mathbf{w}'^{T} = 0 \\ 1 & \text{otherwise} \end{cases}$$

We will choose $\mathbf{v}$ and $\mathbf{w}$ as random Gaussian vectors. Note that the resulting branching program is also a permutation BP.

6.2.4 Modification of CVW Obfuscation

We give here how to modify the CVW obfuscation to be applicable to the resulting permutation BPs of the above transform. We also assume that the index length $h = (\lambda + 1) \cdot \ell$ and the index-to-input function satisfies $\mathsf{inp}(i) = (i \mod \ell)$ as in the CVW obfuscation. We also assume that the BP is $(\lambda + 1)$-input repetition BP as in the original construction. The changed parts are written in red. Note that the targeted BPs have width $2w$. Thus we set $t := (2w + 2n\ell) \cdot n$.

- Sample bundling matrices $\{\mathbf{R}_{i,b} \in \mathbb{Z}^{2n\ell \times 2n\ell}\}_{i\in[h],b\in\{0,1\}}$ such that $(\mathbf{1}^{1\times 2\ell} \otimes \mathbf{I}^{n\times n}) \cdot \mathbf{R}_{\mathbf{x}'} \cdot (\mathbf{1}^{2\ell\times 1} \otimes \mathbf{I}^{n\times n}) = \mathbf{0} \iff \mathbf{x}' \in \bar{\omega}(\{0,1\}^\ell)$ for all $\mathbf{x}' \in \{0,1\}^h$. More precisely, $\mathbf{R}_{i,b}$ is a block diagonal matrix $\mathsf{diag}(\mathbf{R}_{i,b}^{(1)}, \mathbf{R}_{i,b}^{(2)}, \cdots, \mathbf{R}_{i,b}^{(\ell)})$. Each $\mathbf{R}_{i,b}^{(k)} \in \mathbb{Z}^{2n\times 2n}$ is one of the following three cases.

$$\mathbf{R}_{i,b}^{(k)} = \begin{cases} \mathbf{I}^{2n\times 2n} & \text{if } \mathsf{inp}(i) \neq k \\ \begin{pmatrix} \tilde{\mathbf{R}}_{i,b}^{(k)} & \\ & \mathbf{I}^{n\times n} \end{pmatrix}, \tilde{\mathbf{R}}_{i,b}^{(k)} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n} & \text{if } \mathsf{inp}(i) = k \text{ and } i \leqslant \lambda\ell \\ \begin{pmatrix} -\mathbf{I}^{n\times n} & \\ & \prod_{j=0}^{\lambda-1} \tilde{\mathbf{R}}_{k+j\ell,b}^{(k)} \end{pmatrix} & \text{if } \mathsf{inp}(i) = k \text{ and } i > \lambda\ell \end{cases}$$

- Sample matrices $\{\mathbf{S}_{i,b} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{n\times n}\}_{i\in[h],b\in\{0,1\}}$, bookend vectors $\mathbf{v} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{w}$

and $\mathbf{w} \leftarrow \mathcal{D}^{w}_{\mathbb{Z},\sigma}$ and compute

$$\mathbf{J} := ((\mathbf{v}| - \mathbf{v}|\mathbf{1}^{1\times 2n\ell}) \otimes \mathbf{I}^{n\times n}) \in \mathbb{Z}^{n\times t}$$
$$\hat{\mathbf{S}}_{i,b} := \begin{pmatrix} \mathbf{P}_{i,b} \otimes \mathbf{S}_{i,b} & \\ & \mathbf{R}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \in \mathbb{Z}^{t\times t}$$
$$\mathbf{L} := ((\mathbf{w}|\mathbf{w}|\mathbf{1}^{1\times 2n\ell})^{T} \otimes \mathbf{I}^{n\times n}) \in \mathbb{Z}^{t\times n}$$

- Sample $(\mathbf{A}_i, \tau_i) \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ for $0 \leqslant i \leqslant h-1$, $\mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{n\times n})$, $\{\mathbf{E}_{i,b} \leftarrow \mathcal{D}^{t\times m}_{\mathbb{Z},\sigma}\}_{i\in[h-1],b\in\{0,1\}}$ and $\{\mathbf{E}_{h,b} \leftarrow \mathcal{D}^{t\times n}_{\mathbb{Z},\sigma}\}_{b\in\{0,1\}}$.
- Run Sample algorithms to obtain

$$\mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m} \leftarrow \mathsf{Sample}(\mathbf{A}_{i-1}, \tau_{i-1}, \hat{\mathbf{S}}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma) \text{ for } 1 \leqslant i \leqslant h-1,$$
$$\mathbf{D}_{h,b} \in \mathbb{Z}^{m\times n} \leftarrow \mathsf{Sample}(\mathbf{A}_{h-1}, \tau_{h-1}, \hat{\mathbf{S}}_{h,b} \cdot \mathbf{L} \cdot \mathbf{A}_h + \mathbf{E}_{h,b}, \sigma).$$

- Define $\mathbf{A}_{\mathbf{J}}$ as a matrix $\mathbf{J} \cdot \mathbf{A}_0 \in \mathbb{Z}^{n\times m}$ and outputs matrices

$$\left\{\mathsf{inp}, \mathbf{A}_{\mathbf{J}}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}\right\}.$$

We omit the procedure and correctness of evaluation that are almost the same as the original one.

6.2.5 Assumptions of lattice preimage sampling

In this section we provide the experimental results of Assumption 1. Our experiments are built upon the preimage sampling algorithm in the [HHSSD17b], an implementation of BP obfuscation [HHSSD17a].[‡] The results imply that

[‡]We also verify the correctness of the attack itself for [HHSSD17a], but with *large entry* BPs. It requires very large number of samples (say 2^{20} but polynomially many)

Parameters			Experiments		Expected
#products	m	$\log_2 \sigma_x^2$	$\log_2 S^2$	$E[X^4]/\sigma^4$	$\log_2 \sigma^2$
2	2191	34.9	80.8	2.937	80.8
2	2771	35.2	81.4	2.702	81.7
2	3352	35.4	82.4	2.677	82.5
3	2771	35.2	128.7	3.025	128.4
4	3352	35.4	177.0	2.900	176.8
5	3932	35.6	225.9	3.068	225.9
7	5621	36.1	328.1	3.210	327.5

Table 6.2: Experiment results on statistical value of preimage sampling. #products stands for the number of producted preimage matrices, σ_x^2 the variance of preimage sampling, S^2 the sample variance, $E[X^4]/\sigma^4$ the sample kurtosis and σ^2 the expected variance. Every experiment is done using 100 samples. The expected variance is computed under the assumption on independency of D's. Every expected kurtosis assuming independency of D's is about 3.

the variance and kurtosis move almost the same as one assumed independency, the correctness of attack only requires much relaxed assumption.

6.2.6 Useful Tools for Computing the Variances

We introduce useful lemmas to help our computation. We note that we consider the random matrix A whose entries are independent.

Lemma 6.2.1. *Let $A = (A_{i,j})$ be a $n \times n$ random matrix where $A_{i,t}$ and $A_{j,t}$ are independent for every $1 \leqslant i < j \leqslant n$ and $1 \leqslant t \leqslant n$. and $X = [X_1, X_2, \cdots, X_n]$ a n-dimensional random vector which is independent to*

to verify the attack with binary entry BPs, which is not easy to experiment because the obfuscation/evaluation of [HHSSD17a] takes long time (say few minutes to obtain one evaluation).

A. Assume that the following conditions for all distinct $i, j, k, l \in [n]$:

$$E[X_i] = 0,\ E[X_i \cdot X_j] = 0,\ E[X_i^3 \cdot X_j] = 0,$$
$$E[X_i^2 \cdot X_j \cdot X_k] = 0,\ \textit{and}\ E[X_i \cdot X_j \cdot X_k \cdot X_l] = 0.$$

Then, a n-dimensional random vector $Y = [Y_1, Y_2, \cdots, Y_n] = A \cdot X$ also satisfies the similar constraints

$$E[Y_i] = 0,\ E[Y_i \cdot Y_j] = 0,\ E[Y_i^3 \cdot Y_j] = 0,$$
$$E[Y_i^2 \cdot Y_j \cdot Y_k] = 0,\ \textit{and}\ E[Y_i \cdot Y_j \cdot Y_k \cdot Y_l] = 0.$$

for all distinct $i, j, k, l \in [n]$.

Proof.

$$\begin{aligned}
E[Y_i \cdot Y_j] &= E\left[\sum_{t=1}^{n}\sum_{s=1}^{n} A_{i,t} \cdot X_t \cdot A_{j,s} \cdot X_s\right] \\
&= \sum_{t=1}^{n}\sum_{s=1}^{n} E[A_{i,t} \cdot X_t \cdot A_{j,s} \cdot X_s] \\
&= \sum_{1 \leqslant t,s \leqslant n, t \neq s} E[A_{i,t} \cdot A_{j,s}] \cdot E[X_t \cdot X_s] + \sum_{t=1}^{n} E[A_{i,t}] \cdot E[A_{j,t}] \cdot E[X_t \cdot X_t] \\
&= 0
\end{aligned}$$

□

Lemma 6.2.2. *Let $\{A_i = (A_i^{j,k})\}_{1 \leqslant i \leqslant t}$ be $n \times n$ random matrices where*

- *$A_i^{j,k}$ follow Gaussian distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ for all $1 \leqslant j, k \leqslant n$ and $1 \leqslant i \leqslant t$,*

- $A_i^{j,s}$ *and* $A_i^{k,s}$ *are independent for every* $1 \leqslant j < k \leqslant n$, $1 \leqslant s \leqslant n$ *and* $1 \leqslant i \leqslant t$,
- $A_1^{i_1,j_1}, \cdots, A_t^{i_t,j_t}$ *are mutually (entrywise) independent for every* $1 \leqslant i_k, j_k \leqslant n$ *for all* k

and $X = (X_{i,j}) = \prod_{k=1}^{t} A_k$ $n \times n$ *random matrix. For all* $i, j, k \in [n]$, *it holds that*

$$
\begin{aligned}
&E[X_{i,j}] = 0,\ Var[X_{i,j}] = n^{t-1} \cdot (\sigma^2)^t, \\
&E[X_{i,j}^4] = 3\left(n(n+2)\right)^{t-1} \cdot (\sigma^2)^{2t}, \\
&E[X_{i,j}^2 \cdot X_{k,j}^2] = \left(n(n+2)\right)^{t-1} \cdot (\sigma^2)^{2t}
\end{aligned}
$$

Proof. We apply mathematical induction on t. For $t = 1$, it is clear because of the property of Gaussian distribution.

We assume that the equations hold when $t = s$ and will show that the same results hold for $t = s + 1$. Let $X' = \prod_{i=1}^{s} A_i$ and $Y = A_{s+1} \cdot X'$. Note that all entries of A_i follow Gaussian distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ satisfy the same condition of the lemma. We denote $A_{s+1} = (A_{i,j})$ for brevity and $Y_{i,j} = \sum_{k=1}^{n} A_{i,k} \cdot X_{k,j}$. Note that the results of Lemma 6.2.1 holds for every column of X, which can be shown in the inductively applying Lemma 6.2.1.

1. $E[Y_{i,j}] = 0$ is clear.

2. Since $E[Y_{i,j}] = 0$, $Var[Y_{i,j}]$ is the same to $E[Y_{i,j}^2]$. Note that we can obtain $E[X_{k,j} \cdot X_{l,j}] = 0$ and for $k \neq l$ by applying Lemma 6.2.1 inductively, thus $E[A_{i,k} \cdot X_{k,j} \cdot A_{i,l} \cdot X_{l,j}] = E[A_{i,k} \cdot A_{i,l}] \cdot E[X_{k,j} \cdot X_{l,j}] = 0$

also holds. Now we obtain

$$\begin{aligned} Var[Y_{i,j}] = E[Y_{i,j}^2] &= E\left[(\sum_{k=1}^{n} A_{i,k} \cdot X_{k,j})^2\right] \\ &= E\left[\sum_{k=1}^{n} A_{i,k}^2 \cdot X_{k,j}^2\right] = \sum_{k=1}^{n} E[A_{i,k}^2] \cdot E[X_{k,j}^2] \\ &= n \cdot \sigma^2 \cdot n^{s-1} \cdot (\sigma^2)^s = n^s \cdot (\sigma^2)^{s+1} \end{aligned}$$

The last equality holds by the inductive hypothesis.

3. Note that $E[Y_{i,j}^4] = E[(\sum_{k=1}^{n} A_{i,k} \cdot X_{k,j})^4]$. It holds that, for $k \neq l$,

$$\begin{aligned} &E[(A_{i,k} \cdot X_{k,j})^3 \cdot (A_{i,l} \cdot X_{l,j})] = E[A_{i,k}^3 \cdot A_{i,l}] \cdot E[X_{k,j}^3 \cdot X_{l,j}] = 0 \\ &E[(A_{i,k} \cdot X_{k,j})^2 \cdot (A_{i,l} \cdot X_{l,j}) \cdot (A_{i,m} \cdot X_{m,j})] = 0 \\ &E[(A_{i,k} \cdot X_{k,j}) \cdot (A_{i,l} \cdot X_{l,j}) \cdot (A_{i,m} \cdot X_{m,j}) \cdot (A_{i,u} \cdot X_{u,j})] = 0 \end{aligned}$$

for all for all distinct $k, l, m, u \in \{1, \cdots, n\}$. By the induction hypothesis, it holds that

$$E[A_{i,k}^4 \cdot X_{k,j}^4] = E[A_{i,k}^4] \cdot E[X_{k,j}^4] = 3\sigma^4 \cdot 3(n(n+2))^{s-1} \cdot (\sigma^2)^{2s}.$$

Therefore, we conclude that

$$E[(\sum_{k=1}^{n} A_{i,k} \cdot X_{k,j})^4] = 3(n(n+2))^s \cdot (\sigma^2)^{2(s+1)}.$$

4. Note that $E[Y_{i,j}^2 \cdot Y_{k,j}^2] = E[(\sum_{m=1}^{n} A_{i,m} \cdot X_{m,j})^2 \cdot (\sum_{u=1}^{n} A_{k,u} \cdot X_{u,j})^2]$.

Then we obtain the similar result as follows:

$$
\begin{aligned}
&E[(\sum_{m=1}^{n} A_{i,m} \cdot X_{m,j})^2 \cdot (\sum_{u=1}^{n} A_{i,u} \cdot X_{u,j})^2] \\
&= E\left[(\sum_{m=1}^{n} A_{i,m}^2 \cdot X_{m,j}^2) \cdot (\sum_{u=1}^{n} A_{k,u}^2 \cdot X_{u,j}^2) \right] \\
&= \sum_{u=1}^{n} \sum_{m=1}^{n} E[A_{i,m}^2 \cdot A_{k,u}^2] \cdot E[X_{m,j}^2 \cdot X_{u,j}^2] = (n(n+2))^s \cdot (\sigma^2)^{2(s+1)}.
\end{aligned}
$$

□

Lemma 6.2.3. *Let $A = (A_{i,j})$ be a $n \times m$ random matrix whose entries satisfy $E[A_{i,j}] = 0$, $E[A_{i,j}^2] = \sigma_1^2$ and $E[A_{i,j}^4] \leqslant C\sigma_1^4$ for all $i \in [n], j \in [m]$ with some constant C, where the entries of A need not to be independent. Let $v = [v_1, \cdots, v_n]$ and $w = [w_1, \cdots, w_m]$ be n-dimensional random vectors whose entries are mutually independent and follow the Gaussian distribution $\mathcal{D}_{\mathbb{Z},\sigma_2}$. If the entries of A are independent to the entries of v and w, then $Y = v \cdot A \cdot w^T$ satisfies the following condition:*

$$E[Y] = 0,\ E[Y^2] = nm \cdot \sigma_1^2 \cdot \sigma_2^4,\ E[Y^4] \leqslant (nm)^4 \cdot (C\sigma_1^4) \cdot (3\sigma_2^4)^2.$$

Proof. Note that $Y = \sum_{j=1}^{m} \sum_{i=1}^{n} v_i \cdot A_{i,j} \cdot w_j$.

1. $E[Y] = E[\sum_{j=1}^{m} \sum_{i=1}^{n} v_i \cdot A_{i,j} \cdot w_j] = \sum_{j=1}^{m} \sum_{i=1}^{n} E[v_i]E[A_{i,j}]E[w_j] = 0.$
2. For all $i, k \in [n], j, l \in [m]$ satisfy $(i,j) \neq (k,l)$, $E[(v_i \cdot A_{i,j} \cdot w_j) \cdot (v_k \cdot A_{k,l} \cdot w_l)] = E[v_i \cdot v_k]E[A_{i,j} \cdot A_{k,l}]E[w_j \cdot w_l] = 0$ since one of $E[v_i \cdot v_k]$

or $E[w_j \cdot w_l]$ is zero. Then it holds that

$$E[Y^2] = E[(\sum_{j=1}^{m}\sum_{i=1}^{n} v_i \cdot A_{i,j} \cdot w_j)^2] = E[\sum_{j=1}^{m}\sum_{i=1}^{n} v_i^2 \cdot A_{i,j}^2 \cdot w_j^2]$$
$$= \sum_{j=1}^{m}\sum_{i=1}^{n} E[v_i^2]E[A_{i,j}^2]E[w_j^2] = nm \cdot \sigma_1^2 \cdot \sigma_2^4.$$

3. By the Cauchy-Schwarz Inequality, it holds

$$E[Y^4] = E[(\sum_{j=1}^{m}\sum_{i=1}^{n} v_i \cdot A_{i,j} \cdot w_j)^4] \leqslant E[(nm)^3 \cdot (\sum_{j=1}^{m}\sum_{i=1}^{n} v_i^4 \cdot A_{i,j}^4 \cdot w_j^4)]$$
$$= (nm)^3 \cdot \sum_{j=1}^{m}\sum_{i=1}^{n} E[v_i^4]E[A_{i,j}^4]E[w_j^4] \leqslant (nm)^4 \cdot (C\sigma_1^4) \cdot (3\sigma_2^4)^2.$$

□

6.2.7 Analysis of CVW Obfuscation

In this section, we describe how to prove the Lemmas in Section 4.3.2. We use the same notation as in Section 4.3. We re-use or abuse the some notations for the different proof for the convenience of the writing. Fix a $\mathbf{x}$ satisfying $\mathcal{O}(\mathbf{P})(\mathbf{x}) = \mathbf{0}$.

Note that the appeared random matrices are of the form

$$(Z_{1,1}^{(\mathbf{P})})_j = \mathbf{J} \cdot \prod_{i=1}^{j} \hat{S}_{i,x_i}^{(\mathbf{P})} \cdot E_{j+1,x_{j+1}}^{(\mathbf{P})} \cdot \prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{P})},$$

where all random matrices included in $(Z_{1,1}^{(\mathbf{P})})_j$ for each j are mutually independent except the matrices D's. Thus, we are only need to carefully

deal with the product of preimage sampled matrices D's to compute sample variances for each j. This issue is resolved assuming the variance of products of D's and bounds of their kurtosises.

More precisely, by the Assumption 1, a product of the random matrices $\hat{D}_j^{(\mathbf{P})} = \prod_{i=j+2}^{h} D_i^{(\mathbf{P})}$ has the variance $\Theta(m^{h-j-2}(\sigma^2)^{h-j-1})$ and its kurtosis is bounded by $O(\mathsf{poly}(\lambda))$. We denote (possibly polynomial) c_0 by the bound of kurtosises in Assumption 1, and c and d the lower and upper bound of $Var[\hat{D}_k^{(\mathbf{P})}]$ for all k, respectively. In other words, it holds that for all k

$$c \leqslant \frac{Var[\hat{D}_k^{(\mathbf{P})}]}{m^{h-k-2}(\sigma^2)^{h-k-1}} \leqslant d \text{ and } \frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} \leqslant c_0.$$

We also remark that all distributions corresponding to random variables appeared in lemmas except $\left(Z_{1,1}^{(\mathbf{P})}\right)_1$ are the same as regardless of the choice of $\mathbf{P} = \mathbf{M}$ or $\mathbf{N}$, because the matrices of branching programs are all zero except the first matrix. Thus we consider the choice of the branching program only in Lemma 4.3.4, which discusses the random variable $\left(Z_{1,1}^{(\mathbf{P})}\right)_1$.

of Lemma 4.3.1 and 4.3.2. We assume that $\mu_1 < \mu_2$ and it is enough to show the result for $\mathbf{M}$. Note that the random matrix $E_j^{(\mathbf{M})}$ is only (possibly) dependent to $D_j^{(\mathbf{M})}$ and the random variables $(Z_{1,1}^{(\mathbf{M})})_{\mu_1}$ and $(Z_{1,1}^{(\mathbf{M})})_{\mu_2}$ do not contain such random variables at the same time. In addition, $(Z_{1,1}^{(\mathbf{M})})_{\mu_1}$ and $(Z_{1,1}^{(\mathbf{M})})_{\mu_2}$ both contain the random matrix $E_{\mu_1+1}^{(\mathbf{M})}$ whose expectation of each entry is zero. Thus, we obtain the desired result.

Similarly, when we express $(Z_{1,1}^{(\mathbf{M})})_{\mu_1} \cdot (Z_{1,1}^{(\mathbf{M})})_{\mu_2}$ into the polynomials of random variables, then every monomial includes one entry of $E_{\mu_1+1}^{(\mathbf{M})}$ and

does not include the entries of $D^{(\mathbf{M})}_{\mu_1+1}$. Since the expectation of every entry of $E^{(\mathbf{M})}_{\mu_1+1}$ is zero, it completes proof. □

of Lemma 4.3.3. As stated above, it suffice to show the result for $\mathbf{M}$. We define $X^{(\mathbf{M})}_{u,v}$, $Y^{(\mathbf{M})}_{u,v}$ and $(Z^{(\mathbf{N})}_{u,v})_0$ be random variables of the (u,v)-th entry of the random matrix $\prod_{k=2}^{h} D^{(\mathbf{M})}_{k,x_k}$, $E^{(\mathbf{M})}_{1,x_1} \cdot \prod_{k=2}^{h} D^{(\mathbf{M})}_{k,x_k}$ and $\mathbf{J} \cdot E^{(\mathbf{M})}_{1,x_1} \cdot \prod_{k=2}^{h} D^{(\mathbf{M})}_{k,x_k}$, respectively.

Then, for all $u \in [t], v \in [n]$, all random variables $X^{(\mathbf{M})}_{u,v}$ have the variance $\Theta(m^{h-2}(\sigma^2)^{h-1})$ by Assumption 1. Moreover, it holds that $E[X^{(\mathbf{M})}_{u,v}] = 0$ and $\dfrac{E[{X^{(\mathbf{M})}_{u,v}}^4]}{Var[X^{(\mathbf{M})}_{u,v}]^2} \leqslant c_0$ by Assumption 1.

Let $E^{(\mathbf{M})}_{u,v}$ be the random variables of (u,v)-th entry of the random matrix $E^{(\mathbf{M})}_{1,x_1}$. Then we can compute variance and kurtosis of $Y^{(\mathbf{M})}_{u,v}$.

$$
\begin{aligned}
E[Y_{u,v}^{(\mathbf{M})}] &= E[\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})}] = \sum_{i=1}^{m} E[E_{u,i}^{(\mathbf{M})}] \cdot E[X_{i,v}^{(\mathbf{M})}] = 0, \\
E[Y_{u,v}^{(\mathbf{M})} \cdot Y_{u',v}^{(\mathbf{M})}] &= E[(\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})}) \cdot (\sum_{j=1}^{m} E_{u',j}^{(\mathbf{M})} \cdot X_{j,v}^{(\mathbf{M})})] \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} E[E_{u,i}^{(\mathbf{M})} \cdot E_{u',j}^{(\mathbf{M})}] \cdot E[X_{i,v}^{(\mathbf{M})} \cdot X_{j,v}^{(\mathbf{M})}] = 0, \\
Var[Y_{u,v}^{(\mathbf{M})}] &= Var[\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})}] \\
&= E[(\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})})^2] - E[\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})}]^2 \\
&= E[(\sum_{i=1}^{m} {E_{u,i}^{(\mathbf{M})}}^2 \cdot {X_{i,v}^{(\mathbf{M})}}^2)] = \Theta(m^{h-1}(\sigma^2)^h), \\
E[{Y_{u,v}^{(\mathbf{M})}}^4] &= E[(\sum_{i=1}^{m} E_{u,i}^{(\mathbf{M})} \cdot X_{i,v}^{(\mathbf{M})})^4] \\
&\leqslant E[m^3 \cdot (\sum_{i=1}^{m} {E_{u,i}^{(\mathbf{M})}}^4 \cdot {X_{i,v}^{(\mathbf{M})}}^4)] \\
&\leqslant m^4 \cdot 3\sigma^4 \cdot c_0 \cdot (m^{h-2}(\sigma^2)^{h-1} \cdot d)^2
\end{aligned}
$$

We observe $(Z_{1,1}^{(\mathbf{M})})_0 = \sum_{i=1}^{w+2n\ell} Y_{n\cdot(i-1)+1,1}^{(\mathbf{M})}$. Then,

$$
\begin{aligned}
Var[(Z_{1,1}^{(\mathbf{M})})_0] &= E\left[\left(\sum_{i=1}^{w+2n\ell} Y_{n\cdot(i-1)+1,1}^{(\mathbf{M})}\right)^2\right] \\
&= E\left[\sum_{i=1}^{w+2n\ell} {Y_{n\cdot(i-1)+1,1}^{(\mathbf{M})}}^2\right] = \Theta((w+2n\ell) \cdot m^{h-1}(\sigma^2)^h).
\end{aligned}
$$

In addition, the upper bound of $E[(Z_{1,1}^{(\mathbf{M})})_0^4]$ can be computed as follows:

$$\begin{aligned} E[(Z_{1,1}^{(\mathbf{M})})_0^4] &= E[(\sum_{i=1}^{w+2n\ell} Y_{n(i-1)+1,1}^{(\mathbf{M})})^4] \\ &\leqslant E[(w+2n\ell)^3 \cdot (\sum_{i=1}^{w+2n\ell} Y_{n(i-1)+1,1}^{(\mathbf{M})^4})] \\ &\leqslant (w+2n\ell)^4 \cdot m^2 \cdot 3c_0 \cdot d^2 \cdot m^{2h-2} \cdot (\sigma^2)^{2h}. \end{aligned}$$

Combining them, we obtain the inequality

$$\left| \frac{E[(Z_{1,1}^{(\mathbf{M})})_0^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_0]^2} \right| \leqslant 3c_0 \cdot m^2(w+2n\ell)^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

All arguments with respect to $\mathbf{N}$ also hold well.

□

of Lemma 4.3.4. Only for this lemma, we give the proof of the two cases; $\mathbf{P} = \mathbf{M}$ and $\mathbf{P} = \mathbf{N}$.

Case 1: $\mathbf{P} = \mathbf{M}$. We now consider a random matrix $\mathbf{J} \cdot \hat{S}_{1,x_1}^{(\mathbf{M})} \cdot E_{2,x_2}^{(\mathbf{M})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{M})}$. Then, this case is a special case of Lemma 4.3.5. Readers refer to the proof of Lemma 4.3.5. Therefore, we can obtain that

$$Var[(Z_{1,1}^{(\mathbf{M})})_1] = \Theta((n^3 \cdot \sigma^2 + (2\ell-1) \cdot n^2) \cdot m^{h-2} \cdot (\sigma^2)^h)$$

and

$$E[(Z_{1,1}^{(\mathbf{M})})_1^4] \leqslant m^2(w+2n\ell)^4 \cdot 9n^8 \cdot 3c_0 \cdot m^{2h-4} \cdot (\sigma^2)^{2(h+1)} \cdot d^2.$$

Combining this we obtain the inequality

$$\left| \frac{E[(Z_{1,1}^{(\mathbf{M})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_1]^2} \right| \leqslant 27c_0 \cdot m^2(w+2n\ell)^4 \cdot n^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Case 2: P = N. For a random matrix $\mathbf{J} \cdot \hat{S}_{1,x_1}^{(\mathbf{N})} \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$, the random variable can be written as

$$\begin{aligned}\mathbf{J} \cdot \hat{S}_{1,x_1}^{(\mathbf{N})} \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})} &= \mathbf{J} \cdot \mathsf{diag}(\mathbf{1}^{w\times w} \otimes S_{1,x_1}^{(\mathbf{N})}, \mathbf{0}^{n^2\times n^2}) \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})} \\ &\quad + \mathbf{J} \cdot \mathsf{diag}(\mathbf{0}^{wn\times wn}, R_{1,x_1}^{(\mathbf{N})} \otimes S_{1,x_1}^{(\mathbf{N})}) E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}.\end{aligned}$$

since $\hat{S}_{1,x_1}^{(\mathbf{N})}$ is $\mathsf{diag}(\mathbf{1}^{w\times w} \otimes S_{1,x_1}^{(\mathbf{N})}, \mathbf{0}^{n^2\times n^2}) + \mathsf{diag}(\mathbf{0}^{wn\times wn}, R_{1,x_1}^{(\mathbf{N})} \otimes S_{1,x_1}^{(\mathbf{N})})$.

By the lemma 6.2.1, the variance of the random matrix $\mathbf{J} \cdot \hat{S}_{1,x_1}^{(\mathbf{N})} \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$ is equal to summation of variances of two above two random matrices.

We only need to compute the variance of the first random matrix $\mathbf{J} \cdot \mathsf{diag}(\mathbf{1}^{w\times w} \otimes S_{1,x_1}^{(\mathbf{N})}, \mathbf{0}^{n^2\times n^2}) \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$; the variance of the latter term is a special case of the Lemma 4.3.5 as the above case.

Let $S_{u,v}^{(\mathbf{N})}$ be the random variables of (u,v)-th entry of the random matrix $S_{1,x_1}^{(\mathbf{N})}$. We define $X_{u,v}^{(\mathbf{N})}$, $Y_{u,v}^{(\mathbf{N})}$ and $(Z_{u,v}^{(\mathbf{N})})_1$ be random variables of the (u,v)-th entry of the random matrix $E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$, $\hat{S}_{1,x_1}^{(\mathbf{N})} \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$ and $\mathbf{J} \cdot \hat{S}_{1,x_1}^{(\mathbf{N})} \cdot E_{2,x_2}^{(\mathbf{N})} \cdot \prod_{k=3}^{h} D_{k,x_k}^{(\mathbf{N})}$, respectively.

Then, we observe $Y_{1,1}^{(\mathbf{N})} = \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i,1}^{(\mathbf{N})} + \cdots + \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i+(w-1)n,1}^{(\mathbf{N})}$ from the definition of Kronecker tensor properties. Then, using Lemma 6.2.1,

we can obtain

$$\begin{aligned} Var[Y_{1,1}^{(\mathbf{N})}] &= E[(\sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i,1}^{(\mathbf{N})} + \cdots + \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i+(w-1)n,1}^{(\mathbf{N})})^2] \\ &= E[\sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})^2} \cdot X_{i,1}^{(\mathbf{N})^2} + \cdots + \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})^2} \cdot X_{i+(w-1)n,1}^{(\mathbf{N})^2}] \\ &= \Theta(wn \cdot (\sigma^2) \cdot m^{h-2} \cdot (\sigma^2)^{h-1}) \\ &= \Theta(wn \cdot m^{h-2} \cdot (\sigma^2)^h). \end{aligned}$$

Moreover, we can calculate an upper bound of $E[Y_{1,1}^{(\mathbf{N})^4}]$ as follows:

$$\begin{aligned} E[Y_{1,1}^{(\mathbf{N})^4}] &= E\left[(\sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i,1}^{(\mathbf{N})} + \cdots + \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})} \cdot X_{i+(w-1)n,1}^{(\mathbf{N})})^4\right] \\ &\leqslant E\left[(wn)^3 \cdot (\sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})^4} \cdot X_{i,1}^{(\mathbf{N})^4} + \cdots + \sum_{i=1}^{n} S_{1,i}^{(\mathbf{N})^4} \cdot X_{i+(w-1)n,1}^{(\mathbf{N})^4})\right] \\ &\leqslant (wn)^4 \cdot 3(\sigma^2)^2 \cdot m^4 \cdot 3c_o \cdot m^{2h-6} \cdot (\sigma^2)^{2(h-1)} \cdot d^2 \\ &= 9c_0 \cdot (wn)^4 m^2 \cdot m^{2h-4} \cdot (\sigma^2)^{2h} \cdot d^2. \end{aligned}$$

Similarly, we can compute $Y_{i,1}^{(\mathbf{N})}$ for $i = 2, \cdots, wn$ in the exactly same way. The equations and inequalities are all equal to the $Y_{1,1}^{(\mathbf{N})}$ case. For $i > wn$, $Y_{i,1}^{(\mathbf{N})}$ is computed as in **Case 1**. In other words, it is the special case $j = 1$ of Lemma 4.3.5 and the result is equal to Case 1 as well. Thus, we omit the how to compute this value.

Note that $Y_{i,1}^{(\mathbf{N})} = Y_{i+(k-1)n,1}^{(\mathbf{N})}$ for all $k = 1, \cdots, wn$. Thus, we obtain the

desired results as follows:

$$\begin{aligned}
Var[(Z_{1,1}^{(\mathbf{N})})_1] &= E[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{N})})^2] \\
&= E[w^2 \cdot Y_{1,1}^{(\mathbf{N})^2} + \sum_{i=w+1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{N})^2}] \\
&= \Theta((w^3 \cdot n + n^3 \cdot \sigma^2 + (2\ell - 1) \cdot n^2) \cdot m^{h-2}(\sigma^2)^h) \\
E[(Z_{1,1}^{(\mathbf{N})})_1^4] &= E\left[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{N})})^4\right] \\
&\leqslant E[(w + 2n\ell)^3 \cdot (\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{N})^4})] \\
&\leqslant (w + 2n\ell)^4 \cdot 27n^8m^2 \cdot c_0 \cdot m^{2h-4} \cdot (\sigma^2)^{2(h+1)} \cdot d^2
\end{aligned}$$

At last, with the two computations, we obtain

$$\left|\frac{E[(Z_{1,1}^{(\mathbf{N})})_1^4]}{Var[(Z_{1,1}^{(\mathbf{N})})_1]^2}\right| \leqslant 27c_0 \cdot (w + 2n\ell)^4 \cdot n^2m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

□

of Lemma 4.3.5. We remark that, as noted in the above proof, this proof works for $j = 1$ as well and this case is used in the above proof. It suffice to prove the case $\mathbf{P} = \mathbf{M}$. Let $1 \leqslant j < \lambda \cdot \ell$ be an integer that $j = \ell \cdot j_1 + j_2$ and $X_{u,v}^{(\mathbf{M})}$ the random variables of the (u, v)-th entry of the random matrix $E_{j+1,x_{j+1}}^{(\mathbf{M})} \prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{M})}$. Then, all random variables $X_{u,v}$ have the variance $\Theta(m^{h-j-1} \cdot (\sigma^2)^{h-j})$, and we have $E[X_{u,v}^{(\mathbf{M})}] = 0$, $E[X_{u,v}^{(\mathbf{M})} \cdot X_{u',v}^{(\mathbf{M})}] = 0$ for distinct u, u' and $E[X_{u,v}^{(\mathbf{M})^4}] \leqslant 3c_0 \cdot m^2 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h-j)} \cdot d^2$ by Assumption 1.

Let $S_{u,v}^{(\mathbf{M})}$ be the random variable of (u,v)-th entry of the random matrix $\prod_{i=1}^{j} S_{i,x_i}^{(\mathbf{M})}$. Then, $Var[S_{u,v}^{(\mathbf{M})}] = n^{j-1} \cdot (\sigma^2)^j$, $E[S_{u,v}^{(\mathbf{M})} \cdot S_{u',v}^{(\mathbf{M})}] = 0$ for distinct u, u' and $E[{S_{u,v}^{(\mathbf{M})}}^4] = 3\{n(n+2)\}^{j-1} \cdot (\sigma^2)^{2j}$ hold.

By the construction of the matrix $R_{i,x_i}^{(\mathbf{M})}$, $\prod_{i=1}^{j} R_{i,x_i}^{(\mathbf{M})}$ is a block-diagonal matrix that consists of $\prod_{i=1}^{j} {R_{i,x_i}^{(k)}}^{(\mathbf{M})} \in \mathbb{Z}^{2n \times 2n}$ for $k \in [\ell]$. Note that $\prod_{i=1}^{j} {R_{i,x_i}^{(k)}}^{(\mathbf{M})}$ is of the form

$$\prod_{i=1}^{j} {R_{i,x_i}^{(k)}}^{(\mathbf{M})} = \begin{cases} \begin{pmatrix} \prod_{i=1}^{j_1+1} {\tilde{R}_{k+\ell(i-1),x_{k+\ell(i-1)}}^{(k)}}^{(\mathbf{M})} & \\ & \mathbf{I}^{n\times n} \end{pmatrix} & \text{if } k = 1, 2, \cdots, j_2 \\ \begin{pmatrix} \prod_{i=1}^{j_1} {\tilde{R}_{k+\ell(i-1),x_{k+\ell(i-1)}}^{(k)}}^{(\mathbf{M})} & \\ & \mathbf{I}^{n\times n} \end{pmatrix} & \text{if } k = j_2+1, \cdots, \ell \end{cases}$$

Let $R_{u,v}^{(\mathbf{M})}$ be the random variables of the (u,v)-th entry of the random matrix upper-left quadrant of $\prod_{i=1}^{j} {R_{i,x_i}^{(1)}}^{(\mathbf{M})}$. Then $Var[{R_{u,v}^{(\mathbf{M})}}^2] = n^{j_1} \cdot (\sigma^2)^{j_1+1}$, $E[R_{u,v}^{(\mathbf{M})} \cdot R_{u',v}^{(\mathbf{M})}] = 0$ and $E[{R_{u,v}^{(\mathbf{M})}}^4] = 3(n(n+2))^{j_1} \cdot (\sigma^2)^{2(j_1+1)}$.

Similarly, we consider the random variables of the (u,v)-th entry of the matrix $\left(\prod_{i=1}^{j} \hat{S}_{i,x_i}^{(\mathbf{M})}\right) \cdot E_{j+1,x_{j+1}}^{(\mathbf{M})} \cdot \left(\prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{M})}\right)$ and denote it by $Y_{u,v}^{(\mathbf{M})}$. Then,

$$\begin{aligned} Var[Y_{1+wn,1}^{(\mathbf{M})}] &= E[(R_{1,1}^{(\mathbf{M})} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})} X_{i+wn,1}^{(\mathbf{M})} + \cdots + R_{1,n}^{(\mathbf{M})} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})} X_{i+n(w+n-1),1}^{(\mathbf{M})})^2] \\ &= \Theta(n^2 \cdot n^{j_1} \cdot (\sigma^2)^{j_1+1} \cdot n^{j-1} \cdot (\sigma^2)^j \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) \\ &= \Theta(n^{j_1+j+1} \cdot (\sigma^2)^{j_1+j+1} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) \end{aligned}$$

because of Lemma 6.2.1. Moreover, it holds that

$$\begin{aligned}
E[Y_{1+wn,1}^{(\mathbf{M})^4}] &= E[(R_{1,1}^{(\mathbf{M})} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})} X_{i+wn,1}^{(\mathbf{M})} + \cdots + R_{1,n}^{(\mathbf{M})} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})} X_{i+n(w+n-1),1}^{(\mathbf{M})})^4] \\
&\leqslant E[(n^2)^3 (R_{1,1}^{(\mathbf{M})^4} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})^4} X_{i+wn,1}^{(\mathbf{M})^4} + \cdots + R_{1,n}^{(\mathbf{M})^4} \sum_{i=1}^{n} S_{1,i}^{(\mathbf{M})^4} X_{i+n(w+n-1),1}^{(\mathbf{M})^4})] \\
&= 27n^8 m^2 \cdot (n(n+2))^{j_1+j-1} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h+j_1+1)} \cdot d^2.
\end{aligned}$$

Therefore, we conclude that

$$\left| \frac{E[Y_{1+wn,1}^{(\mathbf{M})^4}]}{Var[Y_{1+wn,1}^{(\mathbf{M})}]^2} \right| \leqslant 27c_0 \cdot n^4 m^2 \cdot \left(1 + \frac{2}{n}\right)^{j_1+j-1} \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

Similarly, we can compute all variances of $Y_{i,1}$ for each i.

$$Var[Y_{i,1}^{(\mathbf{M})}] = \begin{cases}
0 & \text{if } i \in [wn] \\
\Theta(n^{j_1+j+1} \cdot (\sigma^2)^{j_1+j+1} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) & \text{if } i = a \cdot n^2 + b + w \cdot n \text{ with } a/2 \in \{0\} \cup [j_2 - 1], b \in [n^2] \\
\Theta(n^{j_1+j} \cdot (\sigma^2)^{j_1+j} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) & \text{if } i = a \cdot n^2 + b + w \cdot n \text{ with } a/2 \in \{j_2, \cdots, \ell\}, b \in [n^2] \\
\Theta(n^j \cdot (\sigma^2)^j \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) & \text{otherwise.}
\end{cases}$$

Thus, we can derive upper bounds of $E[{Y_{i,1}^{(\mathbf{M})}}^4]$ as follows:

$$E[{Y_{i,1}^{(\mathbf{M})}}^4] \leqslant \begin{cases} 0 \\ 27n^8m^2 \cdot \{n(n+2)\}^{j_1+j-1} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h+j_1+1)} \cdot d^2 \\ 27n^8m^2 \cdot \{n(n+2)\}^{j_1+j-2} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h+j_1)} \cdot d^2 \\ 9n^4m^2 \cdot \{n(n+2)\}^{j-1} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2h} \cdot d^2 \end{cases}$$

Let $(Z_{u,v}^{(\mathbf{M})})_j$ be random variable of (u,v)-th entry of the matrix $\mathbf{J} \cdot \left(\prod_{i=1}^{j} \hat{S}_{i,x_i}^{(\mathbf{M})}\right) \cdot E_{j+1,x_{j+1}}^{(\mathbf{M})} \cdot \left(\prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{M})}\right)$. Then, we observe $(Z_{1,1}^{(\mathbf{M})})_j = \sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})}$. Since, by Lemma 6.2.1, $E[S_{u,v}^{(\mathbf{M})} \cdot S_{u',v}^{(\mathbf{M})}] = 0$, $E[R_{u,v}^{(\mathbf{M})} \cdot R_{u',v}^{(\mathbf{M})}] = 0$, and $E[X_{u,v}^{(\mathbf{M})} \cdot X_{u',v}^{(\mathbf{M})}] = 0$ hold for all distinct u, u', the equation $E[Y_{u,1}^{(\mathbf{M})} \cdot Y_{v,1}^{(\mathbf{M})}] = 0$ holds for all u, v.

With the similar method, we compute $Var[(Z_{1,1}^{(\mathbf{M})})_j]$ and upper bound of $E[(Z_{1,1}^{(\mathbf{M})})_j^4]$.

$$\begin{aligned} Var[(Z_{1,1}^{(\mathbf{M})})_j] &= E[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})})^2] = E[\sum_{i=1}^{w+2n\ell} {Y_{1+(i-1)n,1}^{(\mathbf{M})}}^2] \\ &= \Theta(j_2 n \cdot n^{j_1+j+1} \cdot (\sigma^2)^{j_1+j+1} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j} \\ &+ (\ell - j_2)n \cdot n^{j_1+j} \cdot (\sigma^2)^{j_1+j} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j} \\ &+ \ell n \cdot n^{j} \cdot (\sigma^2)^{j} \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j}) \\ &= \Theta(\left(j_2 n^{j_1+j+2}(\sigma^2)^{j_1+1} + (\ell - j_2)n^{j_1+j+1}(\sigma^2)^{j_1} + \ell n^{j+1}\right) m^{h-j-1}(\sigma^2)^h) \end{aligned}$$

$$
\begin{aligned}
E[(Z_{1,1}^{(\mathbf{M})})_j^4] &= E[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})})^4] \\
&\leqslant E[(w+2n\ell)^3 \cdot (\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})^4})] \\
&\leqslant (w+2n\ell)^3\{j_2 n 27n^8 m^2 (n(n+2))^{j_1+j-1} c_0 m^{2h-2j-2} (\sigma^2)^{2(h+j_1+1)} d^2 \\
&+ (\ell - j_2) n \cdot 27n^8 m^2 \cdot (n(n+2))^{j_1+j-2} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h+j_1)} \cdot d^2 \\
&+ \ell n \cdot 9n^4 m^2 \cdot (n(n+2))^{j-1} \cdot c_0 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2h} \cdot d^2\} \\
&\leqslant (w+2n\ell)^4 \cdot 27n^8 m^2 \cdot (n(n+2))^{j_1+j-1} c_0 m^{2h-2j-2} \cdot (\sigma^2)^{2(h+j_1+1)} \cdot d^2
\end{aligned}
$$

Overall, we obtain

$$
\left| \frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2} \right| \leqslant 27c_0 \cdot (w+2n\ell)^4 \cdot n^2 m^2 \cdot \left(1 + \frac{2}{n}\right)^{j_1+j-1} \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).
$$

All arguments for $\mathbf{N}$ hold as well.

□

of Lemma 4.3.6. Similarly, we also focus on the case $\mathbf{P} = \mathbf{M}$. Let j be an integer that $j > \lambda \cdot \ell$ and $j = \ell \cdot \lambda + j_2$. This proof is very similar to Lemma 4.3.4. The difference only comes from a form of the random matrix $\prod_{i=1}^{j} R_{i,x_i}^{(\mathbf{M})}$. Thus, in this proof, we focus on the form of the matrix. Note that, because of the functionality, the matrices $R_{i,b}^{(\mathbf{M})}$ are completely different for $i \leqslant \lambda \cdot \ell$ and for $i > \lambda \cdot \ell$.

In this case, $\prod_{i=1}^{j} R_{i,x_i}^{(\mathbf{M})}$ is the block diagonal matrix

$$
\prod_{i=1}^{j} R_{i,x_i}^{(\mathbf{M})} = \mathsf{diag}(\prod_{i=1}^{j} R_{i,x_i}^{(1)^{(\mathbf{M})}}, \prod_{i=1}^{j} R_{i,x_i}^{(2)^{(\mathbf{M})}}, \cdots, \prod_{i=1}^{j} R_{i,x_i}^{(\ell)^{(\mathbf{M})}})
$$

where $\prod_{i=1}^{j} R_{i,x_i}^{(k)^{(\mathbf{M})}}$ is of the form

$$\begin{cases} \begin{pmatrix} -\prod_{i=1}^{\lambda} \tilde{R}_{k+\ell(i-1),x_{k+\ell(i-1)}}^{(k)^{(\mathbf{M})}} & \\ & \prod_{i=1}^{\lambda} \tilde{R}_{k+\ell(i-1),x_{k+\ell(i-1)}}^{(k)^{(\mathbf{M})}} \end{pmatrix} & \text{if } k = 1, 2, \cdots, j_2 \\ \begin{pmatrix} \prod_{i=1}^{\lambda} \tilde{R}_{k+\ell(i-1),x_{k+\ell(i-1)}}^{(k)^{(\mathbf{M})}} & \\ & \mathbf{I} \end{pmatrix} & \text{if } k = j_2+1, \cdots, \ell \end{cases}$$

Let $Y_{u,v}^{(\mathbf{M})}$ and $(Z_{u,v}^{(\mathbf{M})})_j$ be random variable of (u,v)-th entry of the matrix $\left(\prod_{i=1}^{j} \hat{S}_{i,x_i}^{(\mathbf{M})}\right) \cdot E_{j+1,x_{j+1}}^{(\mathbf{M})} \cdot \left(\prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{M})}\right)$ and $\mathbf{J} \cdot \left(\prod_{i=1}^{j} \hat{S}_{i,x_i}^{(\mathbf{M})}\right) \cdot E_{j+1,x_{j+1}}^{(\mathbf{M})} \cdot \left(\prod_{k=j+2}^{h} D_{k,x_k}^{(\mathbf{M})}\right)$, respectively.

Similarly, we get

$$\begin{aligned} Var[(Z_{1,1}^{(\mathbf{M})})_j] &= E\left[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})})^2 \right] \\ &= E\left[\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})^2} \right] \\ &= \Theta(\left((\ell+j_2)n^{\lambda+j+1} \cdot (\sigma^2)^{\lambda} + (\ell-j_2)n^{j+1}\right) \cdot m^{h-j-1} \cdot (\sigma^2)^h) \end{aligned}$$

and

$$\begin{aligned} E[(Z_{1,1}^{(\mathbf{M})})_j^4] &= E[(\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})})^4] \\ &\leqslant E[(w+2n\ell)^3 (\sum_{i=1}^{w+2n\ell} Y_{1+(i-1)n,1}^{(\mathbf{M})^4})] \\ &\leqslant (w+2n\ell)^4 27 n^8 m^2 (n(n+2))^{\lambda+j-2} c_0 m^{2h-2j-2} (\sigma^2)^{2(h+\lambda)} d^2 \end{aligned}$$

Then, we have

$$\left|\frac{E[(Z_{1,1}^{(\mathbf{M})})_j^4]}{Var[(Z_{1,1}^{(\mathbf{M})})_j]^2}\right| \leqslant 27c_0(w+2n\ell)^4n^2m^2\left(1+\frac{2}{n}\right)^{\lambda+j-2}\left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

The arguments for $\mathbf{N}$ hold as well.

□

6.2.8 Analysis of BGMZ Obfuscation

In this section, we describe how to proof lemmas in Section 4.4.2. We modify the notation as in the CVW obfuscation case. We replace n', n with n, t. We re-use or abuse the some notations for the different proof for the convenience of the writing. For example, we omit the index j in the main body of the paper. Fix a $\mathbf{x} \in \{0,1\}^\ell$ satisfying $\mathcal{O}(\mathbf{P})(\mathbf{x}) = \mathbf{0}$.

By Assumption 1, a product of the random matrices $\hat{D}_j^{\mathbf{P}} = \prod_{i=j+2}^{h} D_i^{(\mathbf{P})}$ has the variance $\Theta(m^{h-j-2}(\sigma^2)^{h-j-1})$ and $O(\mathsf{poly}(\lambda))$ upper bound of its kurtosises.

More precisely, We denote (possibly polynomial) c_0 by the bound of kurtosises in Assumption 1, and c and d the lower and upper bound of $Var[\hat{D}_k^{(\mathbf{P})}]$ for all k, respectively. In other words, it holds that for all k

$$c \leqslant \frac{Var[\hat{D}_k^{(\mathbf{P})}]}{m^{h-k-2}(\sigma^2)^{h-k-1}} \leqslant d \text{ and } \frac{E[(\hat{D}_k{}^{(\mathbf{P})} - E[\hat{D}_k{}^{(\mathbf{P})}])^4]}{Var[\hat{D}_k{}^{(\mathbf{P})}]^2} \leqslant c_0.$$

We omit the proof of Lemma 4.4.1, 4.4.2 since it is almost the same to the proof of Lemma 4.3.1 and Lemma 4.3.2.

of Lemma 4.4.3. Let $(X_{u,v}^{(\mathbf{M})})$ be random variables of the (u,v)-th entry of the random matrix $E_{\mathbf{x}(1)}^{(\mathbf{M})}\prod_{k=2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})}$. Then, for all $u \in [t], v \in [n]$, all

random variables $X_{u,v}^{(\mathbf{M})}$ have the variance $\Theta(m^{h-1}(\sigma^2)^{h-1} \cdot s^2)$. Moreover, it holds that $E[X_{u,v}^{(\mathbf{M})}] = 0$, $E[X_{u,v}^{(\mathbf{M})} \cdot X_{u',v}^{(\mathbf{M})}] = 0$ for distinct u, u' and $E[X_{u,v}^{(\mathbf{M})^4}] \leqslant 3c_0 \cdot m^2 \cdot m^{2h-2} \cdot (\sigma^2)^{2(h-1)} \cdot (s^2)^2 \cdot d^2$ by Assumption 1.

Similarly, the random variables of the (u, v)-th entry of the random matrix $J^{(\mathbf{M})} \cdot E_{1,\mathbf{x}(1)}^{(\mathbf{M})} \prod_{k=2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})}$ are denoted by $Y_{u,v}^{(\mathbf{M})}$. J is defined by $[J'^{(\mathbf{M})}|\mathbf{I}^{n\times n}]$ and $J'^{(\mathbf{M})} \leftarrow \{0,1\}^{n\times wn}$. Let the random variables of the (u, v)-th entry of the random matrix $J'^{(\mathbf{M})}$ be denoted by $J_{u,v}'^{(\mathbf{M})}$. Then we can observe that $E[J_{u,v}'^{(\mathbf{M})}] = \frac{1}{2}$, $E[J_{u,v}'^{(\mathbf{M})^2}] = \frac{1}{2}$, $E[J_{u,v}'^{(\mathbf{M})^4}] = \frac{1}{2}$ for all u, v.

Since $Y_{1,1}^{(\mathbf{M})} = \sum_{i=1}^{w} J_{1,n\cdot(t-1)+1}'^{(\mathbf{M})} \cdot X_{n\cdot(t-1)+1,1}^{(\mathbf{M})} + X_{wn+1,1}^{(\mathbf{M})}$,

$$\begin{aligned} Var[Y_{1,1}^{(\mathbf{M})}] &= E\left[\left(\sum_{i=1}^{w} J_{1,n\cdot(t-1)+1}'^{(\mathbf{M})} \cdot X_{n\cdot(t-1)+1,1}^{(\mathbf{M})} + X_{wn+1,1}^{(\mathbf{M})}\right)^2\right] \\ &= E\left[\sum_{i=1}^{w} J_{1,n\cdot(t-1)+1}'^{(\mathbf{M})^2} \cdot X_{n\cdot(t-1)+1,1}^{(\mathbf{M})^2} + X_{wn+1,1}^{(\mathbf{M})^2}\right] \\ &= \Theta((\frac{w}{2}+1) \cdot m^{h-1} \cdot (\sigma^2)^{h-1} \cdot s^2). \end{aligned}$$

In addition, the upper bound of $E[Y_{1,1}^{(\mathbf{M})^4}]$ can be computed

$$\begin{aligned} E[Y_{1,1}^{(\mathbf{M})^4}] &= E[(\sum_{i=1}^{w} J_{1,n(t-1)+1}'^{(\mathbf{M})} \cdot X_{n(t-1)+1,1}^{(\mathbf{M})} + X_{wn+1}^{(\mathbf{M})})^4] \\ &\leqslant E[(w+1)^3 \cdot (\sum_{i=1}^{w} J_{1,n(t-1)+1}'^{(\mathbf{M})^4} \cdot X_{n(t-1)+1,1}^{(\mathbf{M})^4} + X_{wn+1}^{(\mathbf{M})^4})] \\ &\leqslant (w+1)^4 \cdot 3c_0 \cdot m^2 \cdot m^{2h-2} \cdot (\sigma^2)^{2(h-1)} \cdot (s^2)^2 \cdot d^2. \end{aligned}$$

Similarly, we can derive the same results for $Y_{u,v}$ for all u, v. The variance of $(Z^{(\mathbf{M})})_0 = v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot E_{1,\mathbf{x}(1)}^{(\mathbf{M})} \prod_{k=2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})} \cdot w'^{(\mathbf{M})^T}$ is computed by

$$\begin{aligned} Var[(Z^{(\mathbf{M})})_0] &= \Theta(nm \cdot (\frac{w}{2}+1) \cdot m^{h-1} \cdot (\sigma^2)^{h-1} \cdot s^2 \cdot \sigma^4) \\ &= \Theta(nm \cdot (\frac{w}{2}+1) \cdot m^{h-1} \cdot (\sigma^2)^{h+1} \cdot s^2) \end{aligned}$$

We also have

$$\begin{aligned} E[(Z^{(\mathbf{M})})_0^4] &\leqslant (nm)^4(w+1)^4 3c_0 m^2 \cdot m^{2h-2} \cdot (\sigma^2)^{2(h-1)} \cdot (s^2)^2 \cdot (3\sigma^4)^2 \cdot d^2 \\ &= 27c_0 \cdot (nm)^4 \cdot (w+1)^4 \cdot m^2 \cdot m^{2h-2} \cdot (\sigma^2)^{2(h+1)} \cdot (s^2)^2 \cdot d^2 \end{aligned}$$

At last the upper bound is computed as

$$\left| \frac{E[(Z^{(\mathbf{M})})_0^4]}{Var[(Z^{(\mathbf{M})})_0]^2} \right| \leqslant 108c_0 \cdot (nm)^2 \cdot (w+1)^2 \cdot m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda)$$

For $\mathbf{N}$, all arguments are exactly same.

□

of Lemma 4.4.4. In this proof we consider the two cases; $\mathbf{P} = \mathbf{M}$ and $\mathbf{P} = \mathbf{N}$.

Case 1: $\mathbf{P} = \mathbf{M}$. Consider a random variable $v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot \hat{S}^{(\mathbf{M})}_{1,\mathbf{x}(1)} \cdot E^{(\mathbf{M})}_{2,\mathbf{x}(2)} \cdot \prod_{k=3}^{h} D^{(\mathbf{M})}_{k,\mathbf{x}(k)} \cdot w'^{(\mathbf{M})^T}$. This is the special case $j = 1$ of Lemma 4.4.5. Readers refer to the proof of Lemma 4.4.5. Based on this the following equation

and inequalities hold:

$$\begin{aligned} Var[(Z^{(\mathbf{M})})_1] &= \Theta(nm \cdot n \cdot m^{h-2} \cdot (\sigma^2)^{h+1} \cdot s^2) \\ E[(Z^{(\mathbf{M})})_1^4] &\leqslant 81c_0 \cdot (nm)^4 \cdot n^4 \cdot m^2 \cdot m^{2h-4} \cdot (\sigma^2)^{2(h+1)} \cdot s^4 \cdot d^2 \\ \left| \frac{E[(Z^{(\mathbf{M})})_1^4]}{Var[(Z^{(\mathbf{M})})_1]^2} \right| &\leqslant 81c_0 \cdot (nm)^2 \cdot n^2 \cdot m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda) \end{aligned}$$

Case 2: P = N. Consider a random variable $v'^{(\mathbf{N})} \cdot J^{(\mathbf{N})} \cdot \hat{S}^{(\mathbf{N})}_{1,\mathbf{x}(1)} \cdot E^{(\mathbf{N})}_{2,\mathbf{x}(2)} \cdot \prod_{k=3}^{h} D^{(\mathbf{N})}_{k,\mathbf{x}(k)} \cdot w'^{(\mathbf{N})^T}$. Let $S^{(\mathbf{N})}_{u,v}$ be random variables of (u,v)-th entry of the random matrix $S^{(\mathbf{N})}_{1,\mathbf{x}(1)}$. Similarly, we define $X^{(\mathbf{N})}_{u,v}$ and $Y^{(\mathbf{N})}_{u,v}$ are random variables of the (u,v)-th entry of the random matrix $E^{(\mathbf{N})}_{2,\mathbf{x}(2)} \prod_{k=3}^{h} D^{(\mathbf{N})}_{k,\mathbf{x}(k)}$ and $J^{(\mathbf{N})} \cdot \hat{S}^{(\mathbf{N})}_{1,\mathbf{x}(1)} \cdot E^{(\mathbf{N})}_{2,\mathbf{x}(2)} \cdot \prod_{k=3}^{h} D^{(\mathbf{N})}_{k,\mathbf{x}(k)}$, respectively. $J^{(\mathbf{N})}$ is defined by $[J'^{(\mathbf{N})}|\mathbf{I}^{n\times n}]$ and $J'^{(\mathbf{N})} \leftarrow \{0,1\}^{n\times wn}$. The random variables of the (u,v)-th entry of the random matrix $J'^{(\mathbf{N})}$ is denoted by $J'^{(\mathbf{N})}_{u,v}$.

Then, we observe

$$Y^{(\mathbf{N})}_{1,1} = \sum_{j=1}^{w} \sum_{i=1+n(j-1)}^{nj} (\sum_{k=1}^{n} J'^{(\mathbf{N})}_{k+n(j-1)} \cdot S^{(\mathbf{N})}_{k,i-n(j-1)}) \cdot X^{(\mathbf{N})}_{i,1} + \sum_{k=1}^{n} S^{(\mathbf{M})}_{1,k} \cdot X^{(\mathbf{M})}_{wn+k,1}.$$

By the Lemma 6.2.1, it holds that

$$
\begin{aligned}
&Var[Y_{1,1}^{(\mathbf{N})}] \\
&= E\left[\left(\sum_{j=1}^{w}\sum_{i=1+n(j-1)}^{nj}(\sum_{k=1}^{n} {J'}_{k+n(j-1)}^{(\mathbf{N})} S_{k,i-n(j-1)}^{(\mathbf{N})})X_{i,1}^{(\mathbf{N})} + \sum_{k=1}^{n} S_{1,k}^{(\mathbf{N})} X_{wn+k,1}^{(\mathbf{N})}\right)^2\right] \\
&= E\left[\sum_{j=1}^{w}\sum_{i=1+n(j-1)}^{nj}(\sum_{k=1}^{n} {J'}_{k+n(j-1)}^{(\mathbf{N})^2} S_{k,i-n(j-1)}^{(\mathbf{N})^2})X_{i,1}^{(\mathbf{N})^2} + \sum_{k=1}^{n} S_{1,k}^{(\mathbf{N})^2} X_{wn+k,1}^{(\mathbf{N})^2}\right] \\
&= \Theta(wn \cdot \left(\frac{n}{2} \cdot \sigma^2\right) \cdot m^{h-2} \cdot (\sigma^2)^{h-2} \cdot s^2 + n \cdot \sigma^2 \cdot m^{h-2} \cdot (\sigma^2)^{h-2} \cdot s^2) \\
&= \Theta(\left(\frac{1}{2} \cdot wn + 1\right) \cdot n \cdot m^{h-2} \cdot (\sigma^2)^{h-1} \cdot s^2)
\end{aligned}
$$

In addition, the upper bound of $E[Y_{1,1}^{(\mathbf{N})^4}]$ can be computed

$$
\begin{aligned}
&E[Y_{1,1}^{(\mathbf{N})^4}] \\
&\quad = E\left[\left(\sum_{j=1}^{w}\sum_{i=1+n(j-1)}^{nj}(\sum_{k=1}^{n} J'^{(\mathbf{N})}_{k+n(j-1)}\cdot S^{(\mathbf{N})}_{k,i-n(j-1)})\cdot X^{(\mathbf{N})}_{i,1}\right)\right. \\
&\quad \left. + \sum_{k=1}^{n} S^{(\mathbf{N})}_{1,k}\cdot\ X^{(\mathbf{N})}_{wn+k,1}\Big)^4\right] \\
&\quad \leqslant E\left[\{(w+1)n\}^3\left(\sum_{j=1}^{w}\sum_{i=1+n(j-1)}^{nj}(\sum_{k=1}^{n} J'^{(\mathbf{N})}_{k+n(j-1)} S^{(\mathbf{N})}_{k,i-n(j-1)})^4 X^{(\mathbf{N})^4}_{i,1}\right)\right. \\
&\quad \left. + \sum_{k=1}^{n} S^{(\mathbf{N})^4}_{1,k}\ X^{(\mathbf{N})^4}_{wn+k,1}\Big)\right] \\
&\quad \leqslant E\left[\{(w+1)n\}^3\left(\sum_{j=1}^{w}\sum_{i=1+n(j-1)}^{nj} n^3(\sum_{k=1}^{n} J'_{k+n(j-1)}{}^{(\mathbf{N})^4} S^{(\mathbf{N})^4}_{k,i-n(j-1)})X^{(\mathbf{N})^4}_{i,1}\right)\right. \\
&\quad \left. + \sum_{k=1}^{n} S^{(\mathbf{N})^4}_{1,k}\ X^{(\mathbf{N})^4}_{wn+k,1}\Big)\right] \\
&\quad \leqslant \{(w+1)n\}^3\{wnn^4(\frac{1}{2}3\sigma^4)3c_0m^2m^{2h-4}(\sigma^2)^{2(h-2)}(s^2)^2d^2 \\
&\quad + n(3\sigma^4)3c_0m^2m^{2h-4}(\sigma^2)^{2(h-2)}(s^2)^2d^2\} \\
&\quad \leqslant 9c_0\cdot\{(w+1)n\}^4\cdot n^4\cdot m^2\cdot(\sigma^2)^{2(h-1)}\cdot(s^2)^2\cdot d^2
\end{aligned}
$$

The same results for $Y^{(\mathbf{N})}_{u,v}$ for all u, v can be shown in the same way. The variance of $(Z^{(\mathbf{N})})_1 = v'^{(\mathbf{N})}\cdot J^{(\mathbf{N})}\cdot \hat{S}_{1,\mathbf{x}(1)}\cdot E^{(\mathbf{N})}_{2,\mathbf{x}(2)}\prod_{k=3}^{h} D^{(\mathbf{N})}_{k,\mathbf{x}(k)}\cdot w'^{(\mathbf{N})^T}$ is computed as follows:

$$\begin{aligned} Var[(Z^{(\mathbf{N})})_1] &= \Theta(nm \cdot \left(\frac{1}{2} \cdot wn + 1\right) \cdot n \cdot m^{h-2} \cdot (\sigma^2)^{h-1} \cdot s^2 \cdot \sigma^4) \\ &= \Theta(nm \cdot \left(\frac{1}{2} \cdot wn + 1\right) \cdot n \cdot m^{h-2} \cdot (\sigma^2)^{h+1} \cdot s^2). \end{aligned}$$

Similarly, we have

$$\begin{aligned} E[(Z^{(\mathbf{N})})_1^4] &\leqslant (nm)^4 9c_0\{(w+1)n\}^4 n^4 m^2 m^{2h-4} (\sigma^2)^{2(h-1)} (s^2)^2 (3\sigma^4)^2 d^2 \\ &= 81c_0(nm)^4\{(w+1)n\}^4 n^4 m^2 m^{2h-4} (\sigma^2)^{2(h+1)} (s^2)^2 d^2 \end{aligned}$$

Then, it holds that

$$\left|\frac{E[(Z^{(\mathbf{N})})_1^4]}{Var[(Z^{(\mathbf{N})})_1]^2}\right| \leqslant 324c_0 \cdot (nm)^2 \cdot \{(w+1)n\}^2 \cdot n^2 \cdot m^2 \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

□

of Lemma 4.4.5. Let $2 \leqslant j \leqslant h-1$ be an integer and $X_{u,v}$ the random variables of the (u,v)-th entry of the random matrix $E^{(\mathbf{M})}_{j+1,\mathbf{x}(j+1)} \prod_{k=j+2}^{h} D^{(\mathbf{M})}_{k,\mathbf{x}(k)}$. All random variables $X^{(\mathbf{M})}_{u,v}$ have the variance $\Theta(m^{h-j-1} \cdot (\sigma^2)^{h-j-1} \cdot s^2)$, and $E[X^{(\mathbf{M})}_{u,v}] = 0$, $E[X^{(\mathbf{M})}_{u,v} \cdot X^{(\mathbf{M})}_{u',v}] = 0$ holds for distinct u, u' and $E[{X^{(\mathbf{M})}_{u,v}}^4] \leqslant 3c_0 \cdot m^2 \cdot m^{2h-2j-2} \cdot (\sigma^2)^{2(h-j-1)} \cdot (s^2)^2 \cdot d^2$ by Assumption 1.

We observe that

$$\prod_{i=1}^{j} \hat{S}^{(\mathbf{M})}_{i,x_i} = \begin{pmatrix} \mathbf{0} & \\ & \prod_{i=1}^{j} S^{(\mathbf{M})}_{i,x_i} \end{pmatrix}.$$

Let $S^{(\mathbf{M})}_{u,v}$ be the random variable of (i,j)-th entry of the random matrix

$\prod_{i=1}^{j} S_{i,x_i}^{(\mathbf{M})}$. Then, it hold that $Var[{S_{u,v}^{(\mathbf{M})}}^2] = n^{j-1} \cdot (\sigma^2)^j$, $E[S_{u,v}^{(\mathbf{M})} \cdot S_{u',v}^{(\mathbf{M})}] = 0$ for distinct u, u' and $E[{S_{u,v}^{(\mathbf{M})}}^4] = 3\{n(n+2)\}^{j-1} \cdot (\sigma^2)^{2j}$.

For a random variable of (u,v)-th entry of the random matrix $J^{(\mathbf{M})} \cdot \left(\prod_{i=1}^{j} \hat{S}_{i,\mathbf{x}(i)}^{(\mathbf{M})}\right) \cdot E_{j+1,\mathbf{x}(j+1)}^{(\mathbf{M})} \cdot \left(\prod_{k=j+2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})}\right)$, we denote it by $Y_{u,v}^{(\mathbf{M})}$. Then a variance of $Y_{u,v}^{(\mathbf{M})}$ can be computed using Lemma 6.2.1.

$$\begin{aligned} Var[Y_{u,v}] &= E\left[\left(\sum_{k=1}^{n} S_{u,k}^{(\mathbf{M})} \cdot X_{wn+k,v}^{(\mathbf{M})}\right)^2\right] = E\left[\sum_{k=1}^{n} {S_{u,k}^{(\mathbf{M})}}^2 \cdot {X_{wn+k,v}^{(\mathbf{M})}}^2\right] \\ &= \Theta(n \cdot n^{j-1} \cdot (\sigma^2)^j \cdot m^{h-j-1} \cdot (\sigma^2)^{h-j-1} \cdot s^2) \\ &= \Theta(n^j \cdot m^{h-j-1} \cdot (\sigma^2)^{h-1} \cdot s^2) \end{aligned}$$

Moreover, it holds that

$$\begin{aligned} E[{Y_{u,v}^{(\mathbf{M})}}^4] &= E\left[\left(\sum_{k=1}^{n} S_{u,k}^{(\mathbf{M})} \cdot X_{wn+k,v}^{(\mathbf{M})}\right)^4\right] \leqslant E\left[n^3 \cdot \left(\sum_{k=1}^{n} {S_{u,k}^{(\mathbf{M})}}^4 \cdot {X_{wn+k,v}^{(\mathbf{M})}}^4\right)\right] \\ &\leqslant n^4 3\{n(n+2)\}^{j-1}(\sigma^2)^{2j} 3c_0 m^2 m^{2h-2j-2}(\sigma^2)^{2(h-j-1)}(s^2)^2 d^2 \\ &= 9c_0 n^4 m^2 \{n(n+2)\}^{j-1} m^{2h-2j-2}(\sigma^2)^{2(h-1)}(s^2)^2 d^2 \end{aligned}$$

By Lemma 6.2.3, we can compute

$$v'^{(\mathbf{M})} \cdot J^{(\mathbf{M})} \cdot \prod_{i=1}^{j} \hat{S}_{i,\mathbf{x}(i)}^{(\mathbf{M})} \cdot E_{j+1,\mathbf{x}(j+1)}^{(\mathbf{M})} \prod_{k=j+2}^{h} D_{k,\mathbf{x}(k)}^{(\mathbf{M})} \cdot {w'^{(\mathbf{M})}}^T$$

which is denoted by $(Z^{(\mathbf{M})})_j$. Then it hold that

$$\begin{aligned} Var[(Z^{(\mathbf{M})})_j] &= \Theta(nm \cdot n^j \cdot m^{h-j-1} \cdot (\sigma^2)^{h-1} \cdot s^2 \cdot \sigma^4) \\ &= \Theta(nm \cdot n^j m^{h-j-1} (\sigma^2)^{h+1} s^2) \\ E[(Z^{(\mathbf{M})})_j^4] &\leqslant 9c_0(nm)^4 n^4 m^2 \{n(n+2)\}^{j-1} m^{2h-2j-2} (\sigma^2)^{2(h-1)} (s^2)^2 (3\sigma^4)^2 d^2 \\ &= 81c_0(nm)^4 n^4 m^2 \{n(n+2)\}^{j-1} m^{2h-2j-2} (\sigma^2)^{2(h+1)} (s^2)^2 d^2. \end{aligned}$$

Overall, it holds that

$$\left| \frac{E[(Z^{(\mathbf{M})})_j^4]}{Var[(Z^{(\mathbf{M})})_j]^2} \right| \leqslant 81c_0(nm)^2 n^2 m^2 \left(1 + \frac{2}{n}\right)^{j-1} \cdot \left(\frac{d}{c}\right)^2 = \mathsf{poly}(\lambda).$$

All arguments hold as well for $\mathbf{N}$.

□

of Lemma 4.4.6. Let $X_{u,v}^{(\mathbf{M})}$ be the random variables of the (u, v)-th entry of the random matrix $\prod_{i=1}^{h-1} B_{i,\mathbf{x}(i)}^{(\mathbf{M})}$. All random variables of entries of $B_{i,\mathbf{x}(i)}^{(\mathbf{M})}$ are mutually independent and follow a uniform distribution $[-\frac{\nu}{2}, \frac{\nu}{2})$. For convenience, we assume random variables follow a uniform distribution $[-\frac{\nu}{2}, \frac{\nu}{2}]$. The complete proof is done by considering the statistical indistinguishability of two uniform random distributions.

We note that the similar computations as in Lemma 6.2.2 hold as well for the uniform distributions. More precisely, for the random variable U_1, U_2 following the uniform distribution over $[-\frac{\nu}{2}, \frac{\nu}{2}]$, it hold that $E[U_1] = 0$, $E[U_1^2] = \frac{1}{12} \cdot \nu(\nu+2)$, $E[U_1^4] = \frac{1}{80} \cdot \nu(\nu+2)\{\nu(\nu+2) - \frac{4}{3}\}$.

Thus, the variance of $X_{u,v}^{(\mathbf{M})}$ is

$$Var[X_{u,v}^{(\mathbf{M})}] = g^{h-2} \cdot \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{h-1}.$$

We also have

$$E[X_{u,v}^{(\mathbf{M})^4}] \leqslant 3 \cdot \{g(g+2)\}^{h-2} \cdot \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{2(h-1)}.$$

By Lemma 6.2.3, we can compute the variance and expectation of quadruple of $b_v^{(\mathbf{M})} \cdot \prod_{i=1}^{h-1} B_{i,\mathbf{x}(i)}^{(\mathbf{M})} \cdot b_w^{(\mathbf{M})^T}$ which is denoted by $(Z^{(\mathbf{M})})_h$.

$$\begin{aligned} Var[(Z^{(\mathbf{M})})_h] &\leqslant g^2 \cdot g^{h-2} \cdot \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{h-1} \cdot \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{2} \\ &= g^h \cdot \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{h+1}, \end{aligned}$$

$$\begin{aligned} E[(Z^{(\mathbf{M})})_h^4] &\leqslant (g^2)^4 3\{g(g+2)\}^{h-2} \left\{ \frac{1}{12}\nu(\nu+2) \right\}^{2(h-1)} \left[3 \left\{ \frac{1}{12}\nu(\nu+2) \right\}^{2} \right]^2 \\ &= 27 \cdot (g^2)^4 \cdot \{g(g+2)\}^{h-2} \left\{ \frac{1}{12} \cdot \nu(\nu+2) \right\}^{2(h+1)}. \end{aligned}$$

As a result, $\left| \frac{E[(Z^{(\mathbf{M})})_h^4]}{Var[(Z^{(\mathbf{M})})_h]^2} \right| \leqslant 27 \cdot (g^2)^2 \cdot \left(1 + \frac{2}{g}\right)^{h-2}$. The same arguments hold as well for $\mathbf{N}$. However, this value is not $\mathsf{poly}(\lambda)$, since g is small constant.

□

Bibliography

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In *Theory of Cryptography Conference*, pages 528–556. Springer, 2015.

[ABD16] Martin Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched ntru assumptions. In *Annual Cryptology Conference*, pages 153–178. Springer, 2016.

[ACLL15] Martin R Albrecht, Catalin Cocis, Fabien Laguillaumie, and Adeline Langlois. Implementing candidate graded encoding schemes from ideal lattices. In *Asiacrypt 2015*, volume 9453. Springer, 2015.

[ADGM17] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over ggh13. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 80. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[AGIS14] Prabhanjan Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding barrington's theorem. In *Proceedings of the 2014 ACM SIGSAC Conference*

on Computer and Communications Security, pages 646–658. ACM, 2014.

[Bar86] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc 1. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 1–5. ACM, 1986.

[BEF+17] Jean-François Biasse, Thomas Espitau, Pierre-Alain Fouque, Alexandre Gélin, and Paul Kirchner. Computing generator in cyclotomic integer rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 60–88. Springer, 2017.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual International Cryptology Conference*, pages 1–18. Springer, 2001.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. *Journal of the ACM (JACM)*, 59(2):6, 2012.

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 221–238. Springer, 2014.

BIBLIOGRAPHY

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of ggh15: Provable security against zeroizing attacks. In *Theory of Cryptography Conference*, pages 544–574. Springer, 2018.

[BMSZ16] Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 764–791. Springer, 2016.

[BR14] Zvika Brakerski and Guy N Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *Theory of Cryptography Conference*, pages 1–25. Springer, 2014.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics, American Mathematical Society*, 324:71–90, 2003.

[BZ17] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.

[CCH+19] Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, and Changmin Lee. Statistical zeroizing attack: Cryptanalysis of candidates of bp obfuscation over ggh15 multilinear map. In *Annual International Cryptology Conference*, pages 253–283. Springer, 2019.

BIBLIOGRAPHY

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrede Lepoint, Hemanta K Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New mmap attacks and their limitations. In *Advances in Cryptology–CRYPTO 2015*, pages 247–266. Springer, 2015.

[CGH17] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 278–307. Springer, 2017.

[CHKL18a] Jung Hee Cheon, Minki Hhan, Jiseung Kim, and Changmin Lee. Cryptanalyses of branching program obfuscations over GGH13 multilinear map from the NTRU problem. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, pages 184–210, 2018.

[CHKL18b] Jung Hee Cheon, Minki Hhan, Jiseung Kim, and Changmin Lee. Cryptanalysis on the HHSS obfuscation arising from absence of safeguards. *IEEE Access*, 6:40096–40104, 2018.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–12. Springer, 2015.

[CHL17] Jung Hee Cheon, Minki Hhan, and Changmin Lee. Cryptanalysis of the overstretched NTRU problem for general modulus polynomial. *IACR Cryptology ePrint Archive*, 2017:484, 2017.

[CJL16] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics*, 19(A):255–266, 2016.

[CLLT16] Jean-Sébastien Coron, Moon Sung Lee, Tancrede Lepoint, and Mehdi Tibouchi. Cryptanalysis of ggh15 multilinear maps. In *Annual Cryptology Conference*, pages 607–628. Springer, 2016.

[CLLT17] Jean-Sébastien Coron, Moon Sung Lee, Tancrede Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over clt13. In *IACR International Workshop on Public Key Cryptography*, pages 41–58. Springer, 2017.

[CLT13] Jean-Sébastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology–CRYPTO 2013*, pages 476–493. Springer, 2013.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. Ggh15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*,

pages 577–607, Cham, 2018. Springer International Publishing.

[DGG+18] Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Pratyay Mukherjee. Obfuscation from low noise multilinear maps. In *International Conference on Cryptology in India*, pages 329–352. Springer, 2018.

[FHHL18] Pooya Farshim, Julia Hesse, Dennis Hofheinz, and Enrique Larraia. Graded encoding schemes from obfuscation. In *IACR International Workshop on Public Key Cryptography*, pages 371–400. Springer, 2018.

[FRS17] Rex Fernando, Peter MR Rasmussen, and Amit Sahai. Preventing clt attacks on obfuscation with linear overhead. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 242–271. Springer, 2017.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Eurocrypt*, volume 7881, pages 1–17. Springer, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 40–49. IEEE Computer Society, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography*, pages 498–527. Springer, 2015.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 467–476. ACM, 2013.

[GMM+16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In *Theory of Cryptography Conference*, pages 241–268. Springer, 2016.

[GN08] Nicolas Gama and Phong Nguyen. Predicting lattice reduction. *Advances in Cryptology–EUROCRYPT 2008*, pages 31–51, 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[HHSSD17a] Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing bp-obfuscation using graph-induced encoding. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 783–798. ACM, 2017.

[HHSSD17b] Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing bp-obfuscation us-

ing graph-induced encoding. `https://github.com/shaih/BPobfus`, 2017.

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of ggh map. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 537–565. Springer, 2016.

[KF17] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched ntru parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–26. Springer, 2017.

[KL19] Jiseung Kim and Changmin Lee. Cryptanalysis of the frs obfuscation. *Submitted*, 2019.

[Lau05] Alan J Laub. *Matrix analysis for scientists and engineers*, volume 91. Siam, 2005.

[LMA+16] Kevin Lewi, Alex J Malozemoff, Daniel Apon, Brent Carmer, Adam Foltzer, Daniel Wagner, David W Archer, Dan Boneh, Jonathan Katz, and Mariana Raykova. 5gen: A framework for prototyping applications using multilinear maps and matrix branching programs. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 981–992. ACM, 2016.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2014.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

[MSW14] Eric Miles, Amit Sahai, and Mor Weiss. Protecting obfuscation against arithmetic attacks. *IACR Cryptology ePrint Archive*, 2014:878, 2014.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. In *Annual Cryptology Conference*, pages 629–658. Springer, 2016.

[MZ18] Fermi Ma and Mark Zhandry. The mmap strikes back: Obfuscation and new multilinear maps immune to clt13 zeroizing attacks. In *Theory of Cryptography Conference*, pages 513–543. Springer, 2018.

[Pel18] Alice Pellet-Mary. Quantum attacks against indistinguishablility obfuscators proved secure in the weak multilinear map model. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, pages 153–183, 2018.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *International Cryptology Conference*, pages 500–517. Springer, 2014.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 2014.

[SZ14] Amit Sahai and Mark Zhandry. Obfuscating low-rank matrix branching programs. *IACR Cryptology ePrint Archive*, 2014:773, 2014.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 439–467. Springer, 2015.

국문초록

기능성이 같은 두 프로그램과, 그 난독화된 프로그램들이 있을 때, 난독화된 프로그램들을 구분할 수 없다면 구분불가능한 난독화라고 한다. 구분불가능한 난독화가 존재한다면, 다중선형함수, 함수암호, 다자간 키교환 등 많은 암호학적인 응용들이 존재하기 때문에, 구분불가능한 난독화를 설계하는 것은 매우 중요한 문제 중 하나이다. 일반적으로, 많은 구분불가능한 난독화들은 다중선형함수 GGH13, CLT13, GGH15를 기반으로 하여 설계되었다.

본 학위 논문에서는, 다중선형함수를 기반으로 하는 난독화 기술들에 대한 안전성 분석을 진행한다. 먼저, GGH13 다중선형함수를 기반으로 하는 모든 난독화 기술들은 현재 파라미터 하에 안전하지 않음을 보인다. 프로그램 변환(program converting), 행렬 제로화 공격(matrix zeroizing attack)이라는 두 가지 새로운 방법을 제안하여 안전성을 분석하였고, 그 결과, 현존하는 모든 GGH13 다중선형함수 기반 난독화 기술이 다항식 시간 내에 NTRU 문제로 환원됨을 보인다.

또한, GGH15 다중선형함수를 기반으로 하는 난독화 기술에 대한 통계적인 공격방법을 제안한다. 통계적 공격방법을 최신 기술인 CVW 난독화, BGMZ 난독화에 적용하여, CVW 난독화가 현재 파라미터에서 안전하지 않음을 보인다. 또한 BGMZ 난독화에서 제안한 대수적 안전성 모델이 이상적인 난독화 기술을 설계하는데 충분하지 않다는 것을 보인다. 실제로, BGMZ 난독화가 안전하지 않은 특이한 파라미터를 제안하여, 우리 공격이 BGMZ에서 제안한 안전성 모델에 해당하지 않음을 보인다.

주요어휘: 안전성 분석, 난독화, 다중선형함수
학번: 2014-21202